

1  
39582  
P. 111  
NASA Contractor Report CR-187129

# Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems

Task III - Advanced Fan Section Grid Generator  
Final Report and Computer Program User's Manual

Andrew J. Crook and Robert A. Delaney  
*Allison Gas Turbine Division of General Motors*  
*Indianapolis, Indiana*

September 1991



Prepared for  
Lewis Research Center  
Under Contract NAS3-25270

NASA (CR-187129) INVESTIGATION OF ADVANCED  
COUNTERROTATION BLADE CONFIGURATION CONCEPTS  
FOR HIGH SPEED TURBOPROP SYSTEMS. TASK III:  
ADVANCED FAN SECTION GRID GENERATOR. FINAL  
REPORT AND COMPUTER PROGRAM USER'S MANUAL

001-52170  
Unclass  
03/07 0039582



## Preface

This manual was prepared by Andrew J. Crook and Robert A. Delaney of the Allison Gas Turbine Division, General Motors Corporation, Indianapolis, IN. The work was performed under NASA Contract NAS3-25270 from February, 1991 to June, 1991. The development of the grid generator was performed by Andrew J. Crook. The Allison program manager for this contract was Robert A. Delaney. The NASA program manager for this contract was Christopher J. Miller.

## Acknowledgements

The authors would like to express their appreciation to the following personnel who contributed to this program:

Dr. Dale W. Thoe for his contribution of the elliptic smoothing procedure for the grid generation program,

Dr. Edward J. Hall for his contribution of the blade coordinate rotation procedure and many figures in this document,

Dr. Christopher J. Miller for his review of the program,

Dr. David P. Miller for the use of his grid generation code, *TIGGERC*,

UNIX is a trademark of AT&T

IRIS is a trademark of Silicon Graphics, Inc.

## TABLE OF CONTENTS

<b>1. SUMMARY</b>	<b>1</b>
<b>2. INTRODUCTION</b>	<b>3</b>
<b>3. GRID GENERATION ALGORITHM</b>	<b>5</b>
3.1 Computational Domain	5
3.2 Meridional Plane Algorithm	7
3.3 Elliptic Smoothing Algorithm	13
3.3.1 Introduction	13
3.3.2 Equations	13
3.3.3 Computational Procedure	14
3.4 Tangential Grid Algorithm	18
<b>4. RESULTS</b>	<b>23</b>
<b>5. CONCLUSIONS</b>	<b>29</b>
<b>REFERENCES</b>	<b>31</b>
<b>APPENDIX A. <i>TIGG3D</i> PROGRAM USER'S MANUAL</b>	<b>33</b>
A.1 <i>TIGG3D</i> PROGRAM INITIALIZATION AND DEMONSTRATION	33

A.1.1	Introduction . . . . .	33
A.1.2	Extracting the Source Files . . . . .	33
A.1.3	Compiling the Source Code . . . . .	34
A.1.4	Running the Graphic-Interactive Test Case . . . . .	36
A.1.5	Running the Batch Code Test Case . . . . .	43
A.2	Parameter/Array Sizes . . . . .	45
A.3	Input File Description: Basic Input Section . . . . .	46
A.4	Input File Description: Block Boundary Input Section . . . . .	60
A.5	Output File Description . . . . .	66
A.6	Subroutine Description . . . . .	67
A.7	<i>TIGG3D</i> : GRAPHIC-INTERACTIVE OPERATION . . . . .	73
A.7.1	Introduction . . . . .	73
A.7.2	Main Menu Options . . . . .	73
A.7.3	Edit Tools Option/Menu . . . . .	75
A.7.4	Station Menu . . . . .	80
A.7.5	Flowpath Menu . . . . .	93
A.7.6	Block Menu . . . . .	101
A.7.7	R-Theta View/Mod . . . . .	109
A.8	<i>TIGG3DB</i> : BATCH OPERATION . . . . .	113



## 1. SUMMARY

A procedure has been developed for generating three-dimensional grids for advanced turbofan engine fan section geometries. This document is a final report outlining the development and techniques used in the procedure. The grid generator is to provide a basic tool upon which aerodynamic analyses for advanced turbofan engine components may be based. The procedure constructs a discrete mesh about engine sections containing the fan stage, an arbitrary number of axisymmetric radial flow splitters, a booster stage, and a bifurcated core/bypass flow duct with guide vanes. The grid generation scheme is consistent with aerodynamic analyses utilizing the average-passage equation system developed by Dr. John Adamczyk at the NASA Lewis Research Center. This flow solution scheme requires a grid generator capable of constructing a series of blade specific grids each having a common axisymmetric mesh, but varying in the circumferential direction according to the geometry of the specific blade row.

The mesh points are distributed in an H-type grid system through the fan and about the fan cowl and engine nacelle, as well as upstream and downstream of the fan. The points are distributed with a transfinite interpolation scheme with axial and radial spacing being user specified. Elliptic smoothing of the grid in the meridional

plane is a post-process option. The grid generation system is configured to require a minimum amount of user input but can also be run interactively for maximum user manipulation and refinement of the grid. This combination should provide the best compromise between code stability, ease of use, and grid quality.



## 2. INTRODUCTION

This document contains the Final Report and the Computer Program User's Manual for the *TIGG3D* (Turbomachinery Interactive Grid Generator 3-Dimensional) program developed by the Allison Gas Turbine Division of the General Motors Corporation under Task III of NASA Contract NAS3-25270. The objective of this task was to develop a three-dimensional H-type grid generator for turbofan engine geometries such as the NASA/GE Energy Efficient Engine seen in Fig. 2.1 .

The first portion of this document contains an outline of the grid generation algorithm. The user can refer to this for some background on the processes and features of the code. The user's manual is attached as an appendix and forms the main portion of the document. The first section of the appendix serves to guide the user through the initial setup of the code, and through a sample grid generation. The central sections of the appendix cover some details on the subroutines, input, and output of the code. A description of the function and operation of the many interactive features of the grid generator is presented in the last sections of the appendix.

This grid generator was developed from a code entitled *TIGGERC* which was developed by Dave Miller of the NASA Lewis Research Center [1] . The user is referred to the documentation for that code for more details on many of the operations.

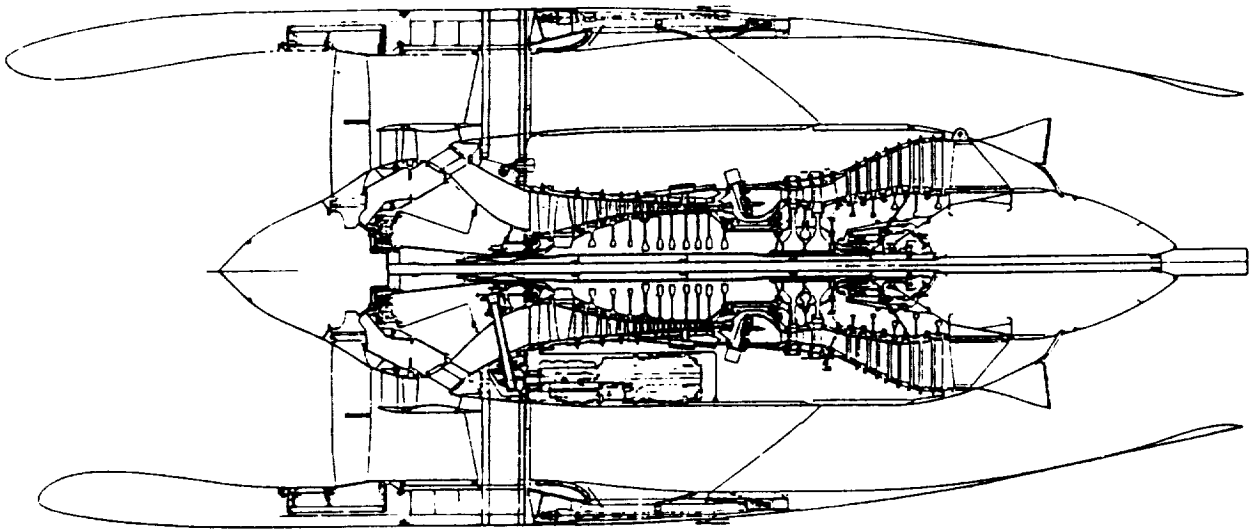


Figure 2.1: NASA/GE Energy Efficient Engine cross section

### 3. GRID GENERATION ALGORITHM

The numerical scheme by which grids are constructed about turbomachinery components is described in this chapter. The geometry and computational domain are briefly described in the first section. Next covered is the algorithm to construct the axisymmetric meridional plane grid through the machine. Once the meridional grid is computed, the option of smoothing the grid with an elliptic PDE is available and a section describes this scheme. Finally, the full three-dimensional grid is constructed and the algorithm for this is the subject of the last section of the chapter.

#### 3.1 Computational Domain

The problem of interest is flow about and through a turbofan geometry in steady operation as shown in Fig. 3.1. Geometric parameters are expressed in a cylindrical coordinate system referenced to the turbofan axis. The axial coordinate lies along the fan rotor axis of rotation, the radial coordinate is perpendicular to the axis of rotation, and the circumferential coordinate sweeps in the counter-clockwise direction when viewed down the axis of rotation (i.e. looking downstream). This coordinate system is illustrated in Fig. 3.1. The hub, cowl, and all other radial splitter contours are assumed to be axisymmetric surfaces (no circumferential variation). Other limitations

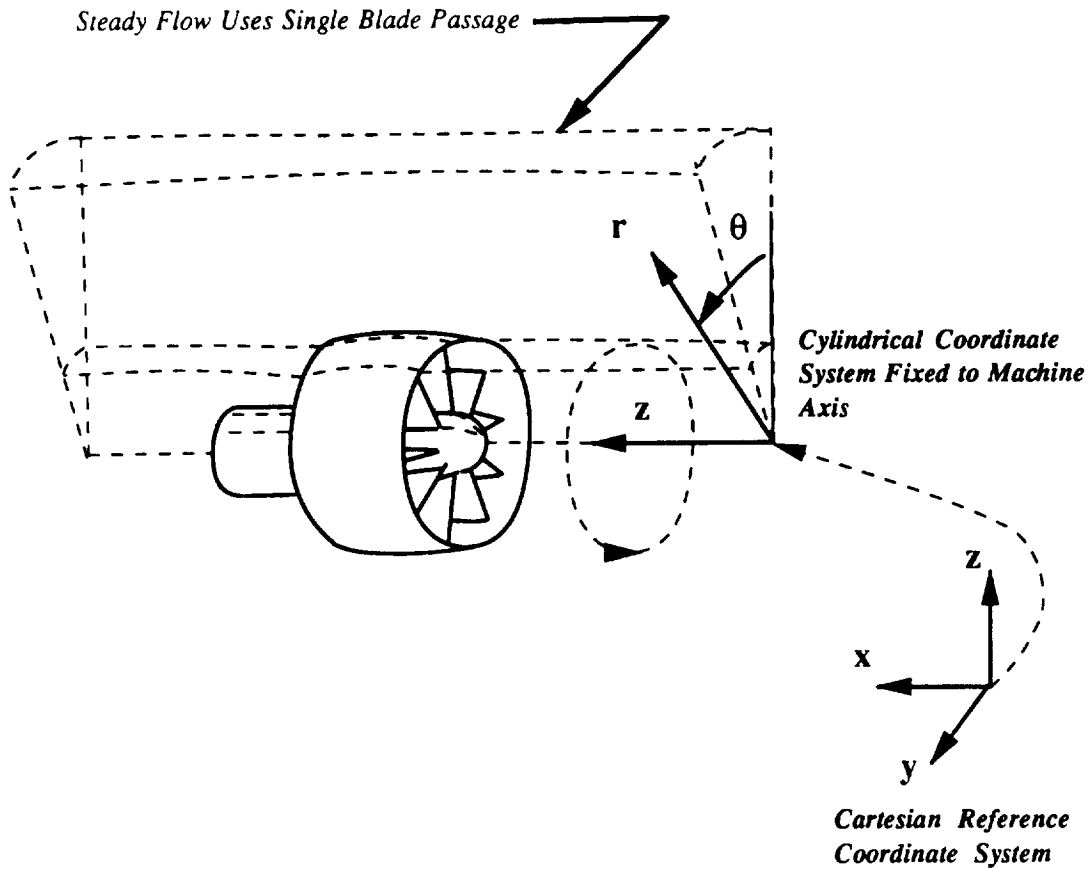


Figure 3.1: Typical turbofan computational domain

are that there must be a small (non-zero) diameter extension running upstream from a spinner stagnation point, and an outer boundary. The spinner extension is needed to allow for the finite volumes inside the “center body” for the radial boundary condition. The outer boundary also must be defined as a constant radius cylinder. For steady flow and a periodic geometry, the computational domain may be reduced to a single blade passage, circumferentially, as shown in Fig. 3.1. For multiple blade row configurations, a single blade passage is used for each blade row and a separate grid is generated for each blade row.

### 3.2 Meridional Plane Algorithm

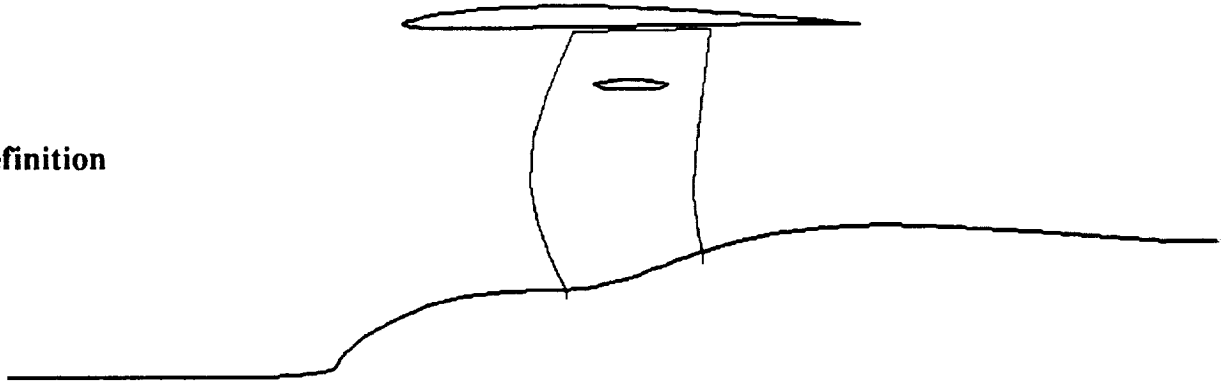
This section describes the procedure to compute a mesh in a two-dimensional meridional plane of the turbomachinery geometry. This portion of the program has undergone only slight modification from the original *TIGGERC* code developed by Dave Miller [1] of NASA Lewis Research Center. The user is referred to [1] for additional details of the algorithm.

Essentially, the meridional plane grid generation takes a series of grid blocks defining a turbomachinery geometry and computes a mesh for each separate grid block. Figure 3.2 shows the steps of this procedure from the given geometry, to the multiple-grid block structural definition, and finally to the mesh generation. The blocks are ordered in rows and columns and there are the same number of blocks for each row.

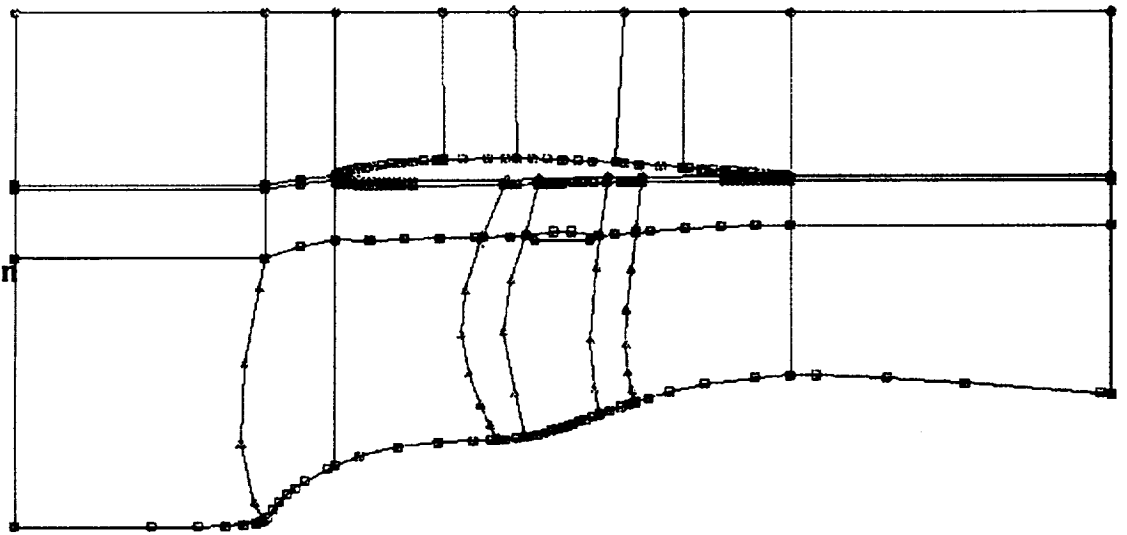
The individual blocks making up the meridional plane structure are defined by four independent line segments. Two of the block boundaries are called flowpath segments (a hub flowpath and shroud flowpath for each individual block) and the other two boundaries are called stations. These block boundaries are illustrated for a sample grid in Fig. 3.3. While often blocks that are adjacent to each other share a common boundary and have the same boundary point locations, they still have individually defined boundaries. This makes it possible to split the hub and shroud flowpaths for neighboring blocks to simulate geometries such as a cowl boundary and the like.

The block boundaries are the key to the grid definition. These boundaries can be input interactively with the mouse controls of the program or by hand. The

**Geometry Definition**



**Grid Block  
Boundary Definition**



**Meridional  
Plane Grid**

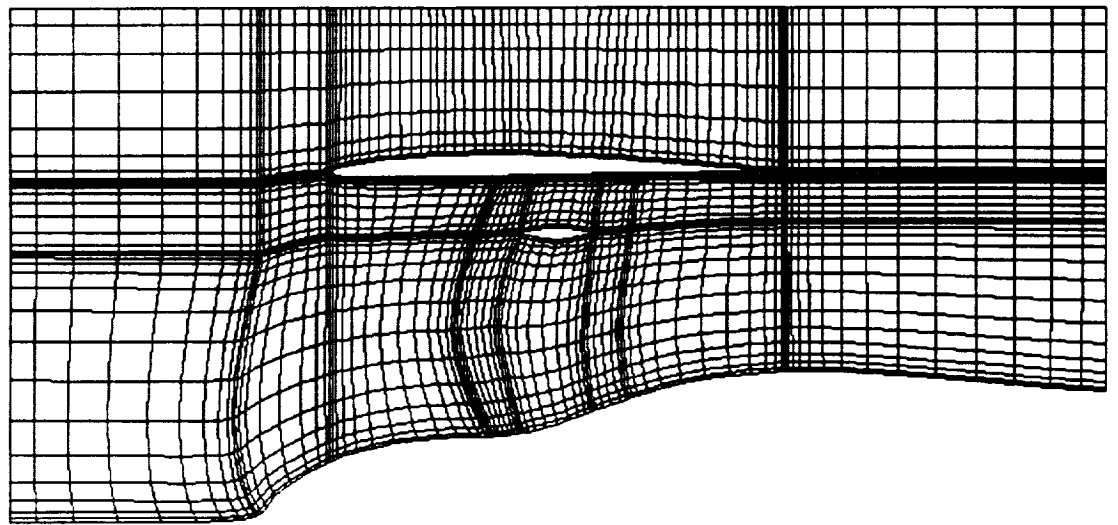


Figure 3.2: Conceptual path of the meridional grid generation scheme

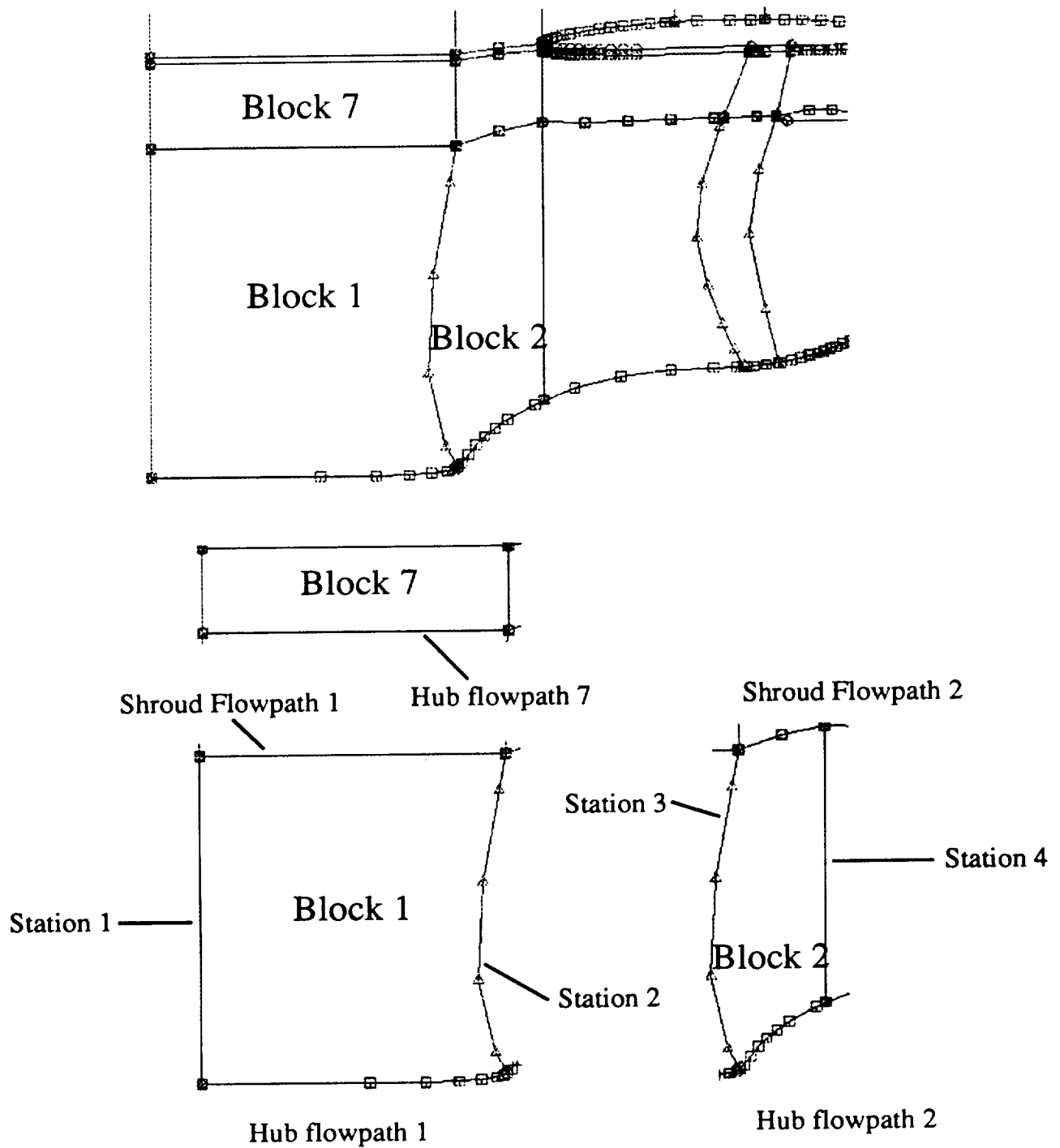


Figure 3.3: Individual blocks and their boundaries for a sample grid

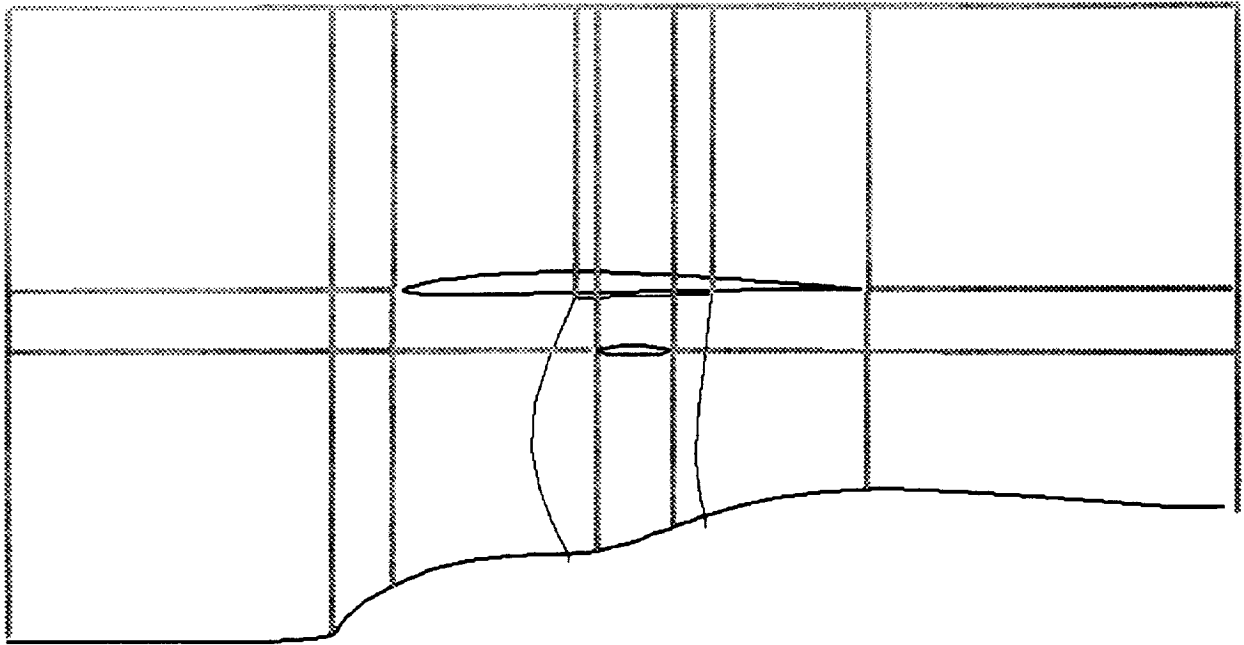


Figure 3.4: Conceptual break up of the domain into block regions at critical points



former method may be inaccurate and both methods can be laborious for complex geometries. To simplify this process, an algorithm has been developed to generate the block boundaries from a general input file which contains only an ordered definition of the pertinent geometry (blade rows and flow splitters). A description of this algorithm follows.

After the input is read, the extreme or pertinent points of the input data are stored and ordered in the axial and radial direction. Examples of extreme or pertinent data points are the leading and trailing edges of the cowl or any other radial flow splitters, spinner nose points, blade leading and trailing edges, and blade tips. Each of these data points defines a new block boundary and a block row or column will be separated at that point (see Fig. 3.4). Once this is done, the radial boundary data is used to create splines which are then used as the definition of all the flowpath lines. Then all the rows and columns are connected to make up the block boundaries.

Once the grid block boundaries are defined, a mesh can be computed for each block. Spacing on each block boundary is defined by a (user specified) minimum length of the first grid cell from that boundary and an algebraic distribution is computed. Then, through a transfinite interpolation process all the interior grid points are computed. By definition this process will produce coincident grid points on the boundary between adjacent blocks if the minimum spacing lengths are the same. This spacing equivalency is taken care of in the *TIGG3D* code to insure continuous mesh definition for the final three-dimensional grid.

While the grid lines may be continuous between neighboring blocks for the meridional plane mesh generator, the slopes of those lines are often quite discontinuous.

The next section will discuss a solution to that undesirable situation.

### 3.3 Elliptic Smoothing Algorithm

#### 3.3.1 Introduction

This section contains a description of the elliptic smoothing technique that is employed in the code *TIGG3D*. The application of elliptic partial differential equations to grid generation problems is a well developed area of computational methods. Some of these techniques are applied in the program *TIGG3D* in order to further process the algebraic grid, which generally is found to possess some unsatisfactory grid properties. Most often these undesired features involve a lack of smoothness and occur at boundaries of the grid where grid blocks abut. The following paragraphs describe briefly the procedures employed in this program.

#### 3.3.2 Equations

The first step in the smoothing process is the assembly of a composite grid formed from the various grid blocks. The 2-D arrays belonging to the blocks are assembled into a single 2-D array. A common boundary between blocks gives rise to double storage of the data corresponding to the boundary. Each point of the composite grid is typed according to whether it is an interior point of a block, a double point of a common shared interior boundary, or a double point of a common outer boundary or metal surface. Points of the latter type are fixed in position.

The assembled grid is viewed as the image of a 2-dimensional rectangle  $\Omega$  :

$$\Omega : 1 \leq \xi \leq M, 1 \leq \eta \leq N;$$

under the transformation

$$\vec{r} = \vec{r}(\xi, \eta).$$

where  $\vec{r}$  denotes the position vector  $\vec{r} = \vec{r}(x, y) = \vec{r}(x(\xi, \eta), y(\xi, \eta))$ . It is well known [3] that a smooth grid with a particular grid clustering will satisfy an elliptic partial differential equation of the form

$$g_{11}\vec{r}_{\xi\xi} + g_{22}\vec{r}_{\eta\eta} - 2g_{12}\vec{r}_{\xi\eta} + g(P\vec{r}_{\xi} + Q\vec{r}_{\eta}) = 0. \quad (1)$$

The coefficients  $g_{11}$ ,  $g_{12}$ , and  $g_{22}$  are metric tensor components,

$$g_{11} = x_{\xi}^2 + y_{\xi}^2, \quad g_{12} = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}, \quad g_{22} = x_{\eta}^2 + y_{\eta}^2, \quad (2)$$

and

$$g = (x_{\xi}y_{\eta} - x_{\eta}y_{\xi})^2. \quad (3)$$

In terms of the physical coordinates  $(x, y)$  the above equation may be written as

$$\nabla^2 \xi = P, \quad \nabla^2 \eta = Q; \quad (4)$$

The functions  $P$  and  $Q$  appearing in the above Poisson equations control the spacing between grid contours;  $P$  controls the degree of clustering in the  $\xi$  direction and  $Q$  controls clustering in the  $\eta$  direction.

### 3.3.3 Computational Procedure

The above concepts are applied to the problem of smoothing the given composite algebraic grid by utilizing the elliptic partial differential equation (1). The partial

derivatives  $\bar{r}_\xi$ ,  $\bar{r}_\eta$ , etc. are evaluated numerically using central differences. Preliminary functions  $\hat{P}$  and  $\hat{Q}$  are then computed from equation (1) . These functions are defined at each point of the composite grid and while definitely not smooth, are accepted as generally describing the sort of clustering desired by the user. The basic idea of the algorithm is to obtain new versions of  $\hat{P}$  and  $\hat{Q}$  for use in the system (1) which are smooth and which still convey the salient clustering information.

If a curve which is a boundary of two grid blocks is to serve as an attractor curve for clustering, say in the  $\xi$  - direction, then the corresponding control function  $\hat{P}$  not only changes sign as the  $\xi$  variable crosses this boundary, but  $\hat{P}$  generally undergoes a rapid change, the magnitude of the variation dependent upon the intensity of the clustering near the curve. As a result, the central differences approximating the partial derivatives in (1) behave wildly across such curves, particularly so at the endpoints . These extreme oscillations introduced into  $\hat{P}$  and  $\hat{Q}$  in this manner are removed by requiring that the ranges of  $\hat{P}$  and  $\hat{Q}$  be restricted to intervals based on the average values of the positive and negative parts of these functions. Without this range clamping feature it was found that the code frequently crashed due to overflows in the Poisson solver. Additionally, because of the manner in which the algebraic grid is generated, there are regions of clustering at points quite distant from the grid boundary or metal surfaces. It is assumed that clustering at such points is not desired, and the functions  $\hat{P}$  and  $\hat{Q}$  are modified at these points in order to minimize their clustering effect. This is accomplished by multiplication of  $\hat{P}$  and  $\hat{Q}$  by scaling functions  $\Phi$  and  $\Psi$  , respectively. The scaling function  $\Phi$  is computed as follows: Let

$d1(\xi) = \text{distance of } \xi \text{ from clustering curve } \xi = \text{constant},$

$d2(\eta) = \text{distance of } \eta \text{ from clustering curve } \eta = \text{constant}.$

The scaling functions are defined by

$$\Phi(\xi, \eta) = (\max(1, 2 - d1(\xi)/D))^2, \quad \Psi(\xi, \eta) = (\max(1, 2 - d2(\eta)/D))^2$$

where D represents a 'cutoff' distance, a parameter which is input by the user.

The final cluster control functions P and Q are obtained from the functions  $\Phi\hat{P}$  and  $\Psi\hat{Q}$  by a smoothing operation. The smoothing is performed by Fourier transform methods using Fast Fourier Transform (FFT) routines. Multiplication of a Fourier transform by a windowing function  $\hat{\sigma}$ , followed by inversion is equivalent to convolving the original function with a smooth function  $\sigma$ , a standard smoothing procedure. If this smoothing operation is denoted by the operator  $\Sigma$ , then the procedure for computing P and Q may be written in the form

$$P = \Sigma\Phi\hat{P}, \quad Q = \Sigma\Psi\hat{Q} \tag{5}$$

The window function  $\sigma$  in use at present is essentially

$$\sigma(\nu) = \max(0, 1 - c\nu^2)$$

where the constant c is determined from a program parameter describing the window size. The present window size in the code is 5. The final step in the grid smoothing procedure is the numerical solution of equation (1) with the coefficients P and Q given by equation (5). The nonlinear system, Eq. (1), is solved in finite difference form using central differences, and using the Gauss-Seidel iteration scheme. Some extra

bookkeeping in the solver is required since the composite grid array contains repeated rows and/or columns corresponding to abutting block boundaries. Also included in the solver is an SOR parameter, which is intended not so much to speed up the gridpoint movement as to slow it down. This was deemed an appropriate inclusion after examining some test cases, and observing the large gridpoint deviations that sometimes occurred during the smoothing computation. The Poisson solver is run for a fixed number of steps, controlled by an input parameter. Running the solver to a convergence criteria based on RMS or absolute error generally produces a grid with severely washed-out clustering in some areas where clustering is desired. A Poisson solver is not an extremely robust scheme, and can be caused to crash easily by use of control functions  $P$  and/or  $Q$  which are widely oscillating or otherwise 'rough'. The above-described scheme provides a means of smoothing an algebraic grid in a rather robust manner.

### 3.4 Tangential Grid Algorithm

Once the meridional plane grid is generated, the axial and radial coordinates of the grid points are established. Orienting the points in the theta direction and computing a tangential grid distribution across the blade pitch is the subject of this section.

The first theta coordinates computed for the meridional grid are for the points defining the actual blade surface. The theta coordinates for the blade row grid points are computed by a line intersection and spline interpolation routine upon the input blade surface data. As stated previously, the three-dimensional grids produced by the **TIGG3D** code are compatible with the average-passage flow solver developed by J.J. Adamczyk at the NASA Lewis Research Center [2] which requires a separate grid for each blade row in the domain. Each grid will have the same axisymmetric or meridional profile but will accurately model only the tangential blade orientation of that grid's particular blade row. Thus, for each independent three-dimensional grid, first the theta coordinates are computed within the region swept out by the subject blade row.

Once the theta coordinates for the blade row block grid points are established, tangential positions for blocks ahead of the leading edge and behind the trailing edge of the blade are found. The tangential distribution from the leading edge is found by computing a line (for each spanwise grid index) with a slope in the axial-tangential plane close to the blade camber angle near the leading edge, transitioning to a slope of zero at a specified grid location. The initial slope from the leading edge is the camber angle between the second and third grid point after the leading edge and averaged in



the radial direction with the grid slice's nearest neighbors. The grid location for zero slope is input by the user to be one, two, or more blocks from the blade leading edge. The same thing is done for the blocks behind the blade trailing edge. A spanwise grid slice displaying a sample tangential distribution is seen in Fig. 3.5.

When the theta coordinates have been computed for the rows of blocks which bound the subject blade row, the rows of blocks above (and below for some blade rows) are found.

After the theta positions of all the grid block points have been found, the tangential distribution of points across the blade pitch is computed. The distribution is algebraic and controlled by a user selected minimum ( $\text{radius} \cdot \Delta \theta$ ) distance for the outer most tangential cells.

In the final compilation of the three-dimensional grid, the coincident points between block boundaries are removed and a continuous mesh is synthesized. If there is a blade tip clearance block and a point discontinuity (because of the step change in blade thickness) the clearance block points are used in the final grid. This results in the blade being abruptly cut down to zero thickness at the blade tip radial index. At all spanwise grid indices where there is a radial split (e.g., a cowl leading edge) the spanwise grid surface is duplicated for the points defining the lower split surface. This is done so that the phantom cells defined in the flow solver will have a grid index associated with them. Figure 3.6 shows this spanwise index relationship.

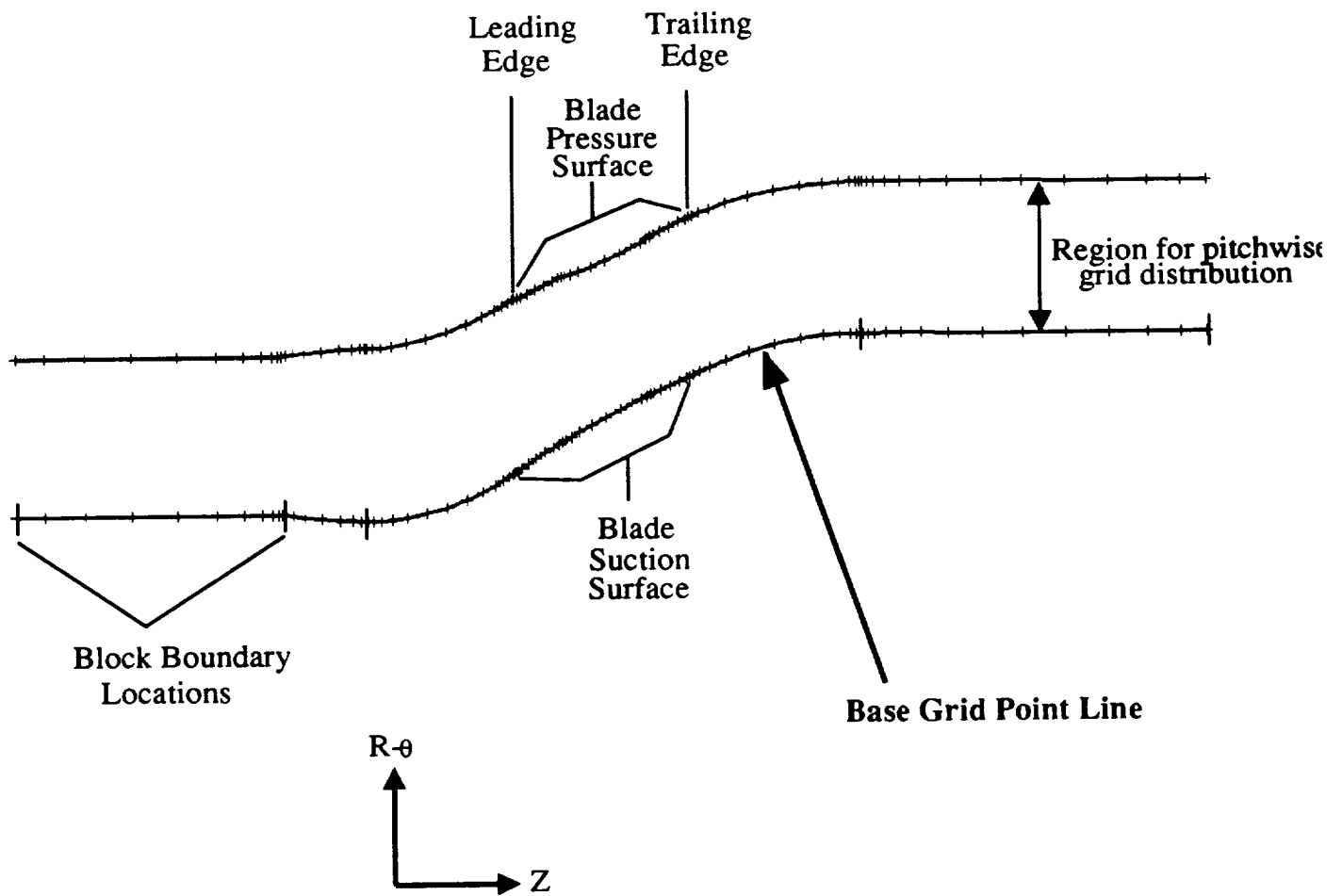


Figure 3.5: Spanwise grid slice in the  $Z$ - $R\theta$  plane

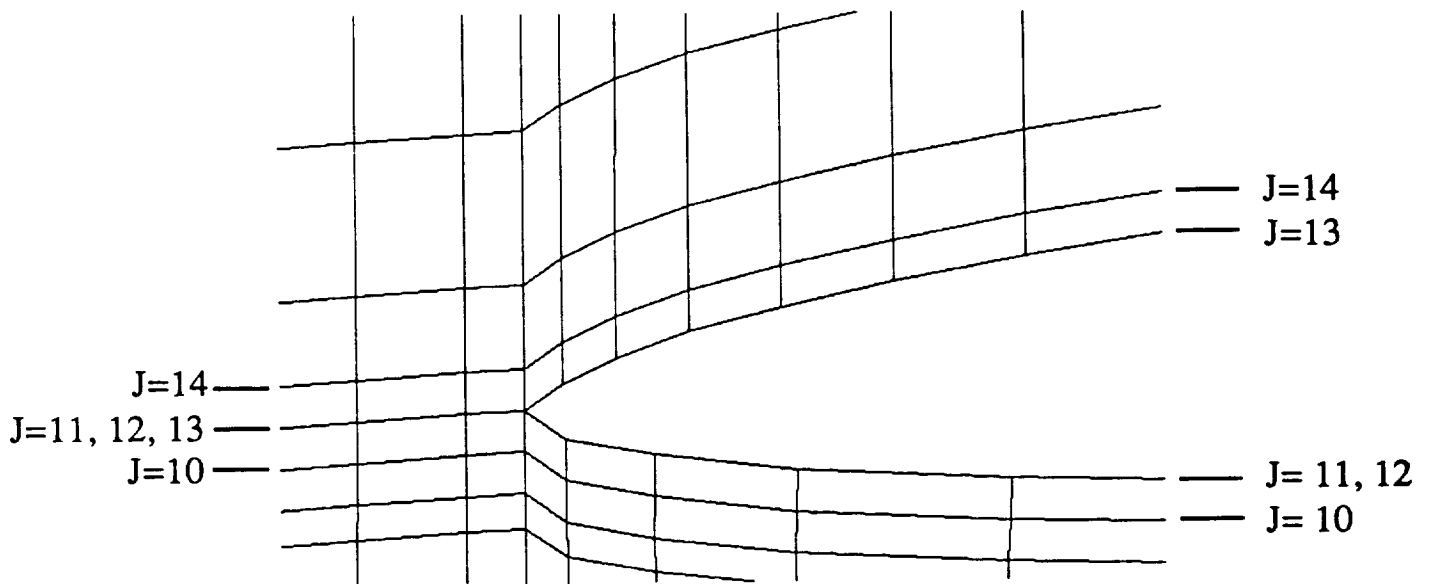


Figure 3.6: Spanwise index numbering around radial splitters



#### 4. RESULTS

A sample grid about a complex turbofan geometry has been generated to demonstrate the capabilities and some features of the *TIGG3D* code. The sample geometry is a mock up of the NASA/GE Energy Efficient Engine with particular emphasis on the fan section. Figure 2.1, seen in the Introduction section, shows some of the details of the fan section.

Geometry data in coordinate pairs ( $z,r$ ) was defined for the hub flowpath, the core/bypass duct splitter, the island splitter, the part-span shroud, and the tip flowpath. Airfoil section data was also generated for all the pertinent blade rows and from that input set, a preliminary grid was generated.

Once the initial grid was generated, the grid point distributions and grid block boundaries in the meridional plane were manipulated a great deal. Figure 4.1 shows a detail of the grid in the fan section. All of the subject blade rows and flow splitters are labeled. This portion of the grid was the first to be manipulated, as it is the section of prime importance. The axial and radial number of points were first selected for the blocks in the blade rows to adequately define those regions for an Euler equation flow solver. Points were clustered about the leading and trailing edges to better resolve the geometry and flowfield in those critical regions. Points were also clustered in the radial direction around the fan rotor blade tip and the radial flow splitters. Finally,

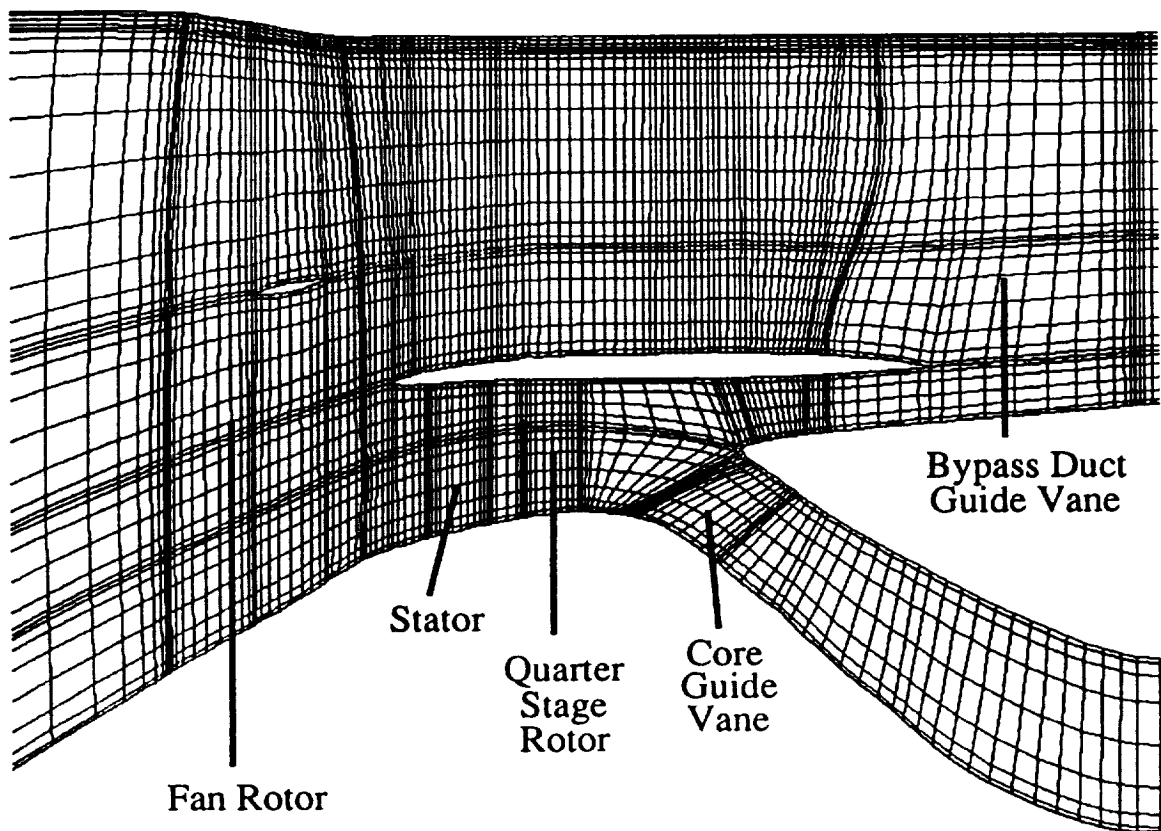


Figure 4.1: Meridional grid about the Energy Efficient Engine fan section

the station boundaries for the blocks over the island splitter were shifted to give a more even distribution of points over the upper splitter surface.

After the meridional grid was refined in the fan section, similar manipulations were done for the entire engine. Figure 4.2 shows the block boundary definitions for the entire grid. An example manipulation is the redefinition of the block boundaries near the cowl leading edge to relieve the high grid shear typically seen there for H-type meshes.

Tangential parameters were then set for the grid and the full three-dimensional grid was generated. A three-dimensional representation of the engine and some grid sections defining the engine is seen in Fig. 4.3 . The green surface is the spinner and hub flowpath and the red surface is the cowl. An axial plane section of the grid from the hub to the outer boundary is seen in white. In this figure, the blades next to the blade grided in yellow have been removed for reasons of clarity. Note that this figure shows a representation of the whole engine and that the actual grid produced is one pitchwise section (from one blade to another), and that two pitchwise sections are seen in white in the figure.

This sample grid for a mock up of the Energy Efficient Engine was made up of 126 grid blocks, 213 axial points, 65 radial points, and 9 tangential points.

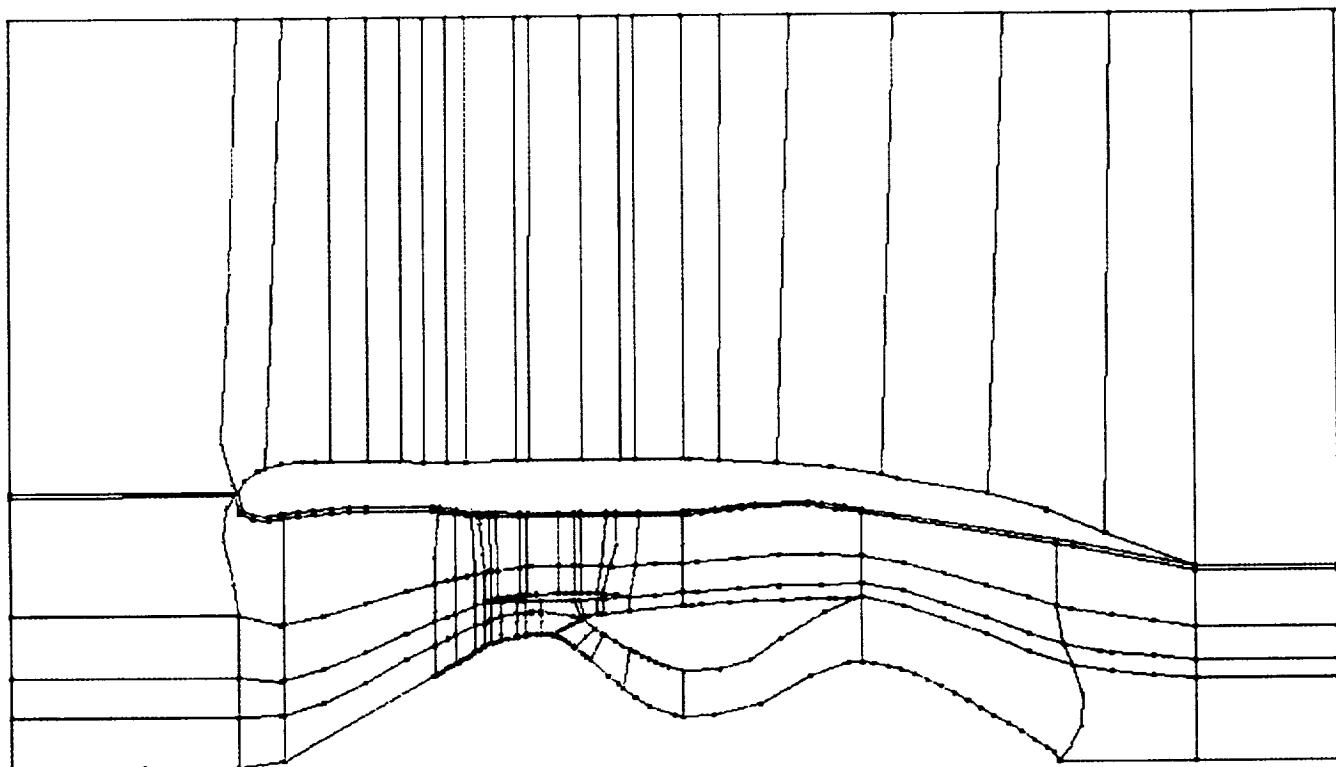


Figure 4.2: Block grid structure about a mock up of the Energy Efficient Engine





Figure 4.3 : 3-D grid about the Energy Efficient Engine highlighting blade region







## 5. CONCLUSIONS

A three-dimensional grid generator for general turbofan geometries has been developed. H-type grids can be generated for multiple blade row and multiple radial flow-split configurations. The grids produced are consistent with the average-passage equation flow solver [2] scheme which requires a series of blade-specific grids (one for each blade row) each having a common axisymmetric mesh.

The grid generator requires only the basic blade and splitter surface geometry as input. The generator is highly user interactive, with graphic menu and mouse driven features which maximize user refinement of the grid. This graphic-interactive version of the generator is machine specific to Silicon Graphics workstations at the time of this report. However, a batch version of the generator which is more system independent is available and uses the same input file as the graphic-interactive version.



## REFERENCES

- [1] Miller, D. P., "TIGGERC - Turbomachinery Interactive Grid Generator Energy Distributor and Restart Code", NASA TM to be published.
- [2] Celestina, M. L., Mulac, R. A., and Adamczyk, J. J., " A Numerical Simulation of the Inviscid Flow Through a Counterrotating Propeller", ASME Paper 86-GT-138, June 1981.
- [3] Thompson, J. E., Warsi, Z.U.A., and Mastin C. W., "Numerical Grid Generation", North-Holland, 1985
- [4] Whipple, D., "BDX-Binary Data Exchange Preliminary Information", NASA-Lewis Research Center, 1989.
- [5] Walatka, P. P., Buning, P. G., Pierce, L., and Elson, P. A., "PLOT3D User's Manual," NASA TM 101067, 1988.





## APPENDIX A. *TIGG3D* PROGRAM USER'S MANUAL

### A.1 *TIGG3D* PROGRAM INITIALIZATION AND DEMONSTRATION

#### A.1.1 Introduction

This section describes the commands necessary to extract the *TIGG3D* source code from the standard distribution and run a complete test case for a grid about a ducted fan with a part span shroud. The standard *TIGG3D* distribution is normally a compressed *tar* file which can be decoded into the various parts by a sequence of commands on any standard UNIX system. The sequence listed below is intended to guide the user through the setup from the standard distribution up to and including a complete demonstration of a graphic-interactive grid generation as well as a grid generation in non-graphic or "batch" mode.

#### A.1.2 Extracting the Source Files

The *TIGG3D* programs are distributed as a compressed *tar* file named

*tigg3d.tar.Z*

While the graphic-interactive code depends on Silicon Graphics commands and is at this time restricted to such machines, it should be possible to extract and run the “batch” code on any standard UNIX system from this distribution file. The first step necessary to extract the *TIGG3D* programs is to uncompress the *tar* file with the command:

**uncompress tigg3d.tar.Z**

This operation essentially replaces the compressed file *tigg3d.tar.Z* with an uncompressed file *tigg3d.tar*.

The next step is to extract the individual files and directories from the *tigg3d.tar* file. The **tar** command will create a subdirectory named *tigg3d* in the current directory, so the user should move the *tigg3d.tar* file to a suitable initial directory. Once the *tar* file is properly placed, the *TIGG3D* distribution may be extracted with the command

**tar xvof tigg3d.tar**

(On some systems **tar xvf tigg3d.tar** may be sufficient.) Execution of the command **ls -l** will verify that the *tigg3d* directory has been created.

### **A.1.3 Compiling the Source Code**

After extracting the source files, the user is naturally interested in compiling the source files for execution. A UNIX-compatible *Make* facility is provided for each of the *TIGG3D* programs. The *Makefile* which governs the compilation process is necessarily machine-dependent and requires that the user select from one of a number of

preconfigured systems. The systems which are immediately available are:

**iris** Silicon Graphics Iris workstation  
**cray** Cray Computer Inc. supercomputer  
**aix** IBM Aix operating system UNIX workstation

If no *system* is specified, then the *iris* system is assumed. The machine dependence of the compilation process is inherently tied to the use of the Scientific DataBase Library routines for binary file input/output.

In order to begin the compilation, it is first necessary to enter the *tigg3d* directory with the command:

**cd tigg3d**

At this point, several files and directories will be available. By entering the command **ls**, a listing of the individual directories can be obtained. The output of the **ls** command will look something like:

**demos/      Makefile      manual/      report/      sdblib/      src/**

A description of each of these listings is given below:

**demos** This directory contains several geometry and flow input files for generating sample runs of the *TIGG3D* codes.

**Makefile** This file is the global *Makefile* for the compiling system.

**manual** This directory contains the *LaTeX* source code for this manual. If *LaTeX* is installed on your system, it is possible to reproduce this document (excluding figures) with the command **latex manual**. The

resulting device independent file *manual.dvi* may then be converted to *PostScript* or previewed on screen through a number of widely available routines.

**report** This directory contains the *LaTeX* source code for the final report outlining the technical details of the *TIGG3D* codes. If *LaTeX* is installed on your system, it is possible to reproduce the final report (excluding figures) with the command **latex finalrep**. The resulting device independent file *finalrep.dvi* may then be converted to *PostScript* or previewed on screen through a number of widely available routines.

**src** This directory contains all the FORTRAN source code for the *TIGG3D* programs *tigg3d* and *tigg3db*.

**sdblib** This directory contains the various machine-dependent files for the Scientific DataBase Library routines [4].

It is now possible to compile the *TIGG3D* codes. By issuing the command

**make system**

where *system* indicates the current computing platform (iris, cray, or aix) described above. From the main directory, issuing the **make** command compiles all source programs. Individual *Makefiles* are also included in each source and tools subdirectory to permit separate compilation of the individual codes.

#### **A.1.4 Running the Graphic-Interactive Test Case**

Once the *make* facility has properly completed compiling the *TIGG3D* source code distribution, it is possible to run the test cases provided with the standard

distribution. It is recommended that the sample cases be run to verify proper compilation and extraction of the *TIGG3D* distribution. Again, the graphic-interactive version of the code can only be run on an Iris workstation so this portion of the demonstration must be on such a machine.

In order to run the demonstration cases, it is necessary to begin in the *demo* directory. At this point, the *demo* directory may be entered by issuing the command

```
cd demo
```

In the *demo* directory, an *ls* command will reveal:

```
demo.input  demo2.input
```

The file *demo.input* is obviously the input file for the demonstration case. The file *demo2.input* is an input file for a two blade row test case. The user may run the code with this second input file to see the small differences in this case. The two blade row case is not outlined explicitly here and the user is referred to the **Graphic-Interactive Operation** section to understand the differences. The grid generation process is launched by issuing the command

```
../src/tigg3d
```

After a title frame appears, a menu graphic and a text window will be displayed on the screen. A representation of the menu graphic is shown in Fig. A.1. This program is mostly mouse driven and menu organized. If shown, the mouse cursor is represented by a red arrow on the screen.

The menu items will be explained in some detail in other sections of this manual. Move the mouse over the portion of the menu graphic with the title **Start/Restart**

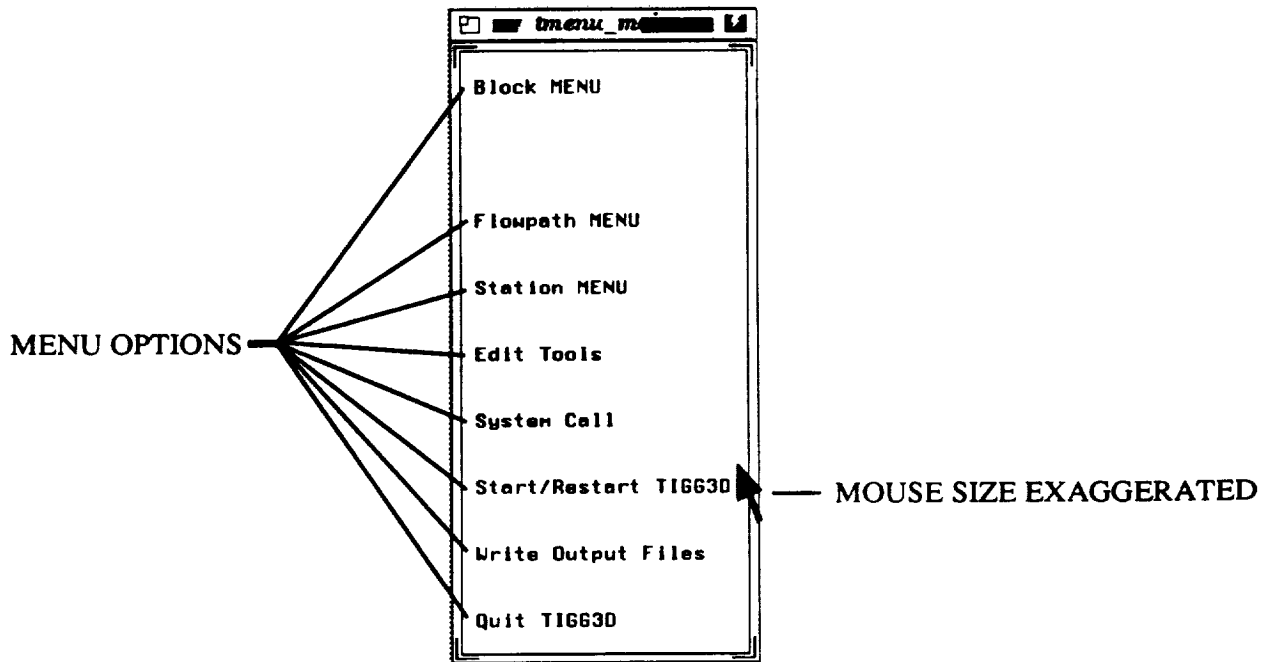


Figure A.1: Main menu graphic

**Tigg3d** and press the right mouse button. The user should notice the color of the menu item change as the mouse moves over it. This should aid in the correct selection of menu options. When the right mouse button is pressed, this executes the menu option to read in the input file, and the user is prompted to enter the name of the input data file into the text window. The mouse should be moved into the text window near the top of the screen and the name

**demo.input**

should be typed in and the enter key pressed.

After this, some messages should pass quickly through the text window and a large graphics window will be produced with a plot of the grid block boundaries as seen in Fig. A.2.

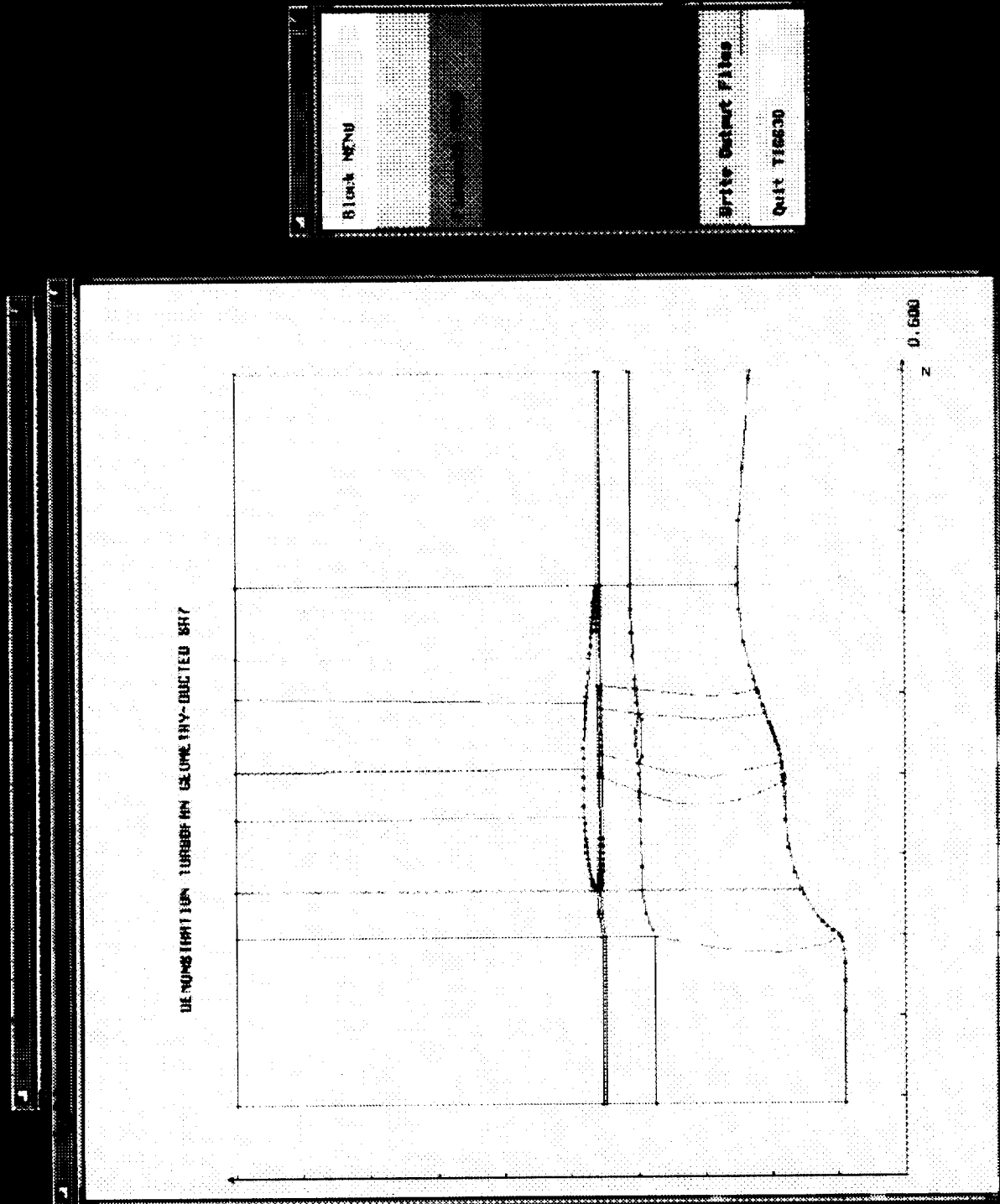


Figure A.2: Workstation screen after input data file is read in by program





At this point the mouse should be moved to the menu item entitled **Block MENU** and the right mouse button pressed. Two things will occur. A mesh will be plotted over the meridional grid blocks in the graphics window and a new menu graphic will be displayed over the original Main Menu. Again, all the menu items will be described in other sections of this manual, but for now, the user should move the mouse to the menu item entitled **Grid Density** and press the right mouse button.

Upon inspection of the grid displayed in the graphics window, the user might note a deficiency in the number of axial points in the region near the leading edge of the cowl. The menu option just selected will address the deficiency. When the **Grid Density** option is selected, the grid density selector graphic will fill the right side of the screen. The screen at this time should look very much like Fig. A.3. Each of the grid blocks has a set of selectors for density and packing at the boundaries. The grid block controlled by a selector is noted under the grid density slider. Moving the mouse into a selector set will highlight the corresponding grid block in blue in the graphics window. For this exercise, move the mouse cursor to the red bar on the density slider for grid block two. Press and hold down the right mouse button and slide the red bar to the right until the number of points, as indicated under the right hand edge of the slider region, changes from five to nine. Now release the right mouse button.

When this has been executed, there should be nine axial points in the grid block instead of five and the meridional mesh should look more satisfactory. To exit this operation the mouse must be in the grid density selector graphic and the left mouse button pressed. When this is done, the grid block menu is displayed on the right side



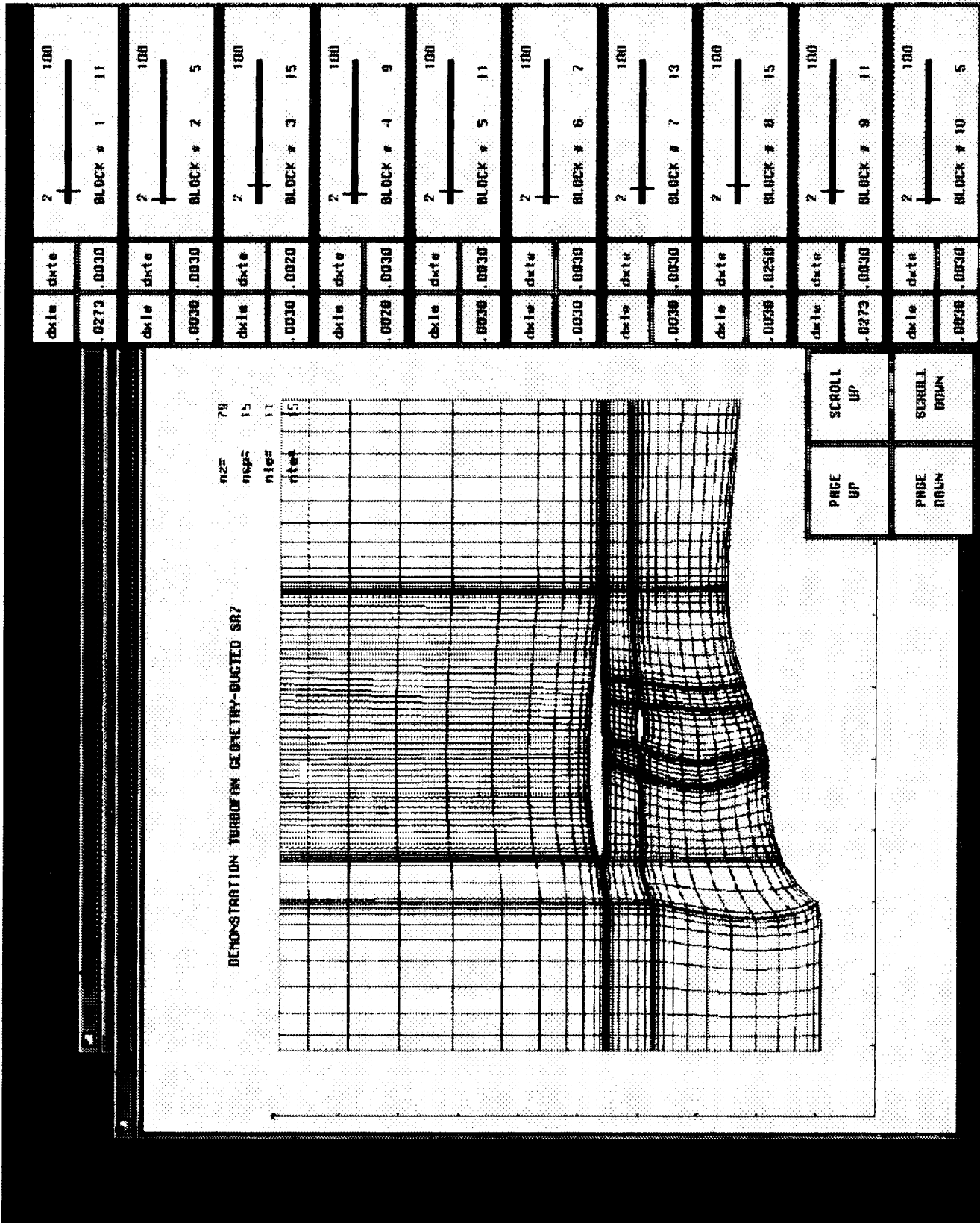


Figure A.3: Workstation screen after Grid Density menu option has been executed



of the screen again.

Next, the mouse should be moved to the menu option entitled **Elliptic Smoothing** and the right mouse button pressed. After a short time a new meridional grid will be plotted in the graphics window.

At this point, move the mouse to the **Theta View/Mod** menu option and press the right mouse button. After a while a plot of the first axial slice of the three-dimensional grid is displayed in the graphics window and a new menu is produced in the upper right corner of the workstation screen. The user should then move the mouse to the new menu option entitled **Plot 3-D Grid** and press the right mouse button. After a significant amount of time (the grid writing process is time consuming) the mouse can be moved to the **Exit Theta MENU** option and the right mouse button pressed. Control of the program is now back to the Block Menu and the mouse should be moved to the **Exit Block MENU** option and again the right mouse button pressed.

The Main Menu should now be displayed on the right of the screen. Before quitting the program, the changes made to grid blocks near the cowl leading edge should be saved. This is done by selecting the option entitled **Write Output File** with the right mouse button. After this, the **Quit Tigg3d** menu option should be selected. Execution of this option generates another menu window like the one shown in Fig. A.4. Select the **Quit Tigg3d** option by pressing the right mouse button.

When this series of steps is completed a three-dimensional grid is generated and written out to a file called

*grid1.bin*



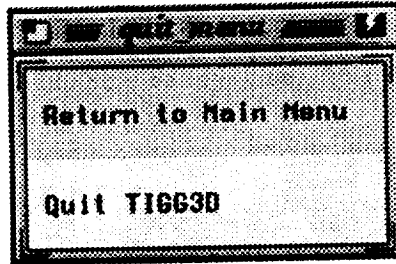


Figure A.4: Quit menu graphic

and a new input file is written out to a file called

*demo.input.new*

The grid is written in a machine independent format compatible with that of a *PLOT3D* [5] multiple-grid binary file.

#### A.1.5 Running the Batch Code Test Case

To run this demonstration case, the user must be in the same directory as was previously used to run the graphic-interactive demonstration case. The input file for this case can be the same as that for the graphic-interactive case or the new input file generated during that demonstration. For this case the original input file will be used. The batch version of *Tigg3d* also uses another input file and that is listed in the directory under the name

*batch.stdin*

To run the batch mode grid generator with the given input file, enter the command:

```
../src/tigg3db < batch.stdin
```

Upon completion of the program, the grid will be written out to the file called

*grid1.bin*

and can be viewed with the *PLOT3D* program.



## A.2 Parameter/Array Sizes

It is necessary to estimate the maximum array sizes and other limit parameters prior to the compilation process. Most of the arrays in the routines of the code are dimensioned according to the `PARAMETER` statements in the file *parameter.gn* which contains the lines:

```
parameter(map=100,mgb=160,nx=200,mrp= 70,mpfp=1002,mrow=12)
parameter(mtp=31, msec=37, mbkpbr=21)
```

These parameters dimension numerous arrays found in the files *tiger.com1* and *m.co1*. While these parameters serve many functions in the code, below is a list of their primary purposes.

- MAP** Maximum number of axial points for each grid block.
- MGB** Maximum number of grid blocks making up the entire grid.
- MRP** Maximum number of radial points for each grid block.
- MPFP** Maximum number of input defining points for the hub flow path and the cowl.
- MROW** Maximum number of blade rows or radial splitters.
- MTP** Maximum number of tangential points defining the grid.
- MSEC** Maximum number of input blade defining sections.
- MBKPBR** Maximum number of blocks that can fall within a single blade row.

Before running the code, the user should estimate the size of the grid and make sure the array sizes are adequate. Errors and peculiar results will occur if the arrays are not properly dimensioned.

When the parameter `MAP` or `MGB` is changed and the graphic-interactive ver-

sion of the code is being used, one additional change is required. The first two lines of the file *common.h* must also be changed to reflect the new value of MAP or MGB. The file *common.h* defines arrays for the C language subroutines needed for the graphic-interactive version of the program.

### A.3 Input File Description: Basic Input Section

A sample input data file for *TIGG3D* is shown in Fig. A.5. The input file consists of an arbitrary number of header lines which is terminated by a line beginning with a "+" character. Beyond this line, the file follows the structured format which is given in Fig. A.5. The structured data format is organized in fields of 10 characters, whether integer or real. Integer data must be right justified in their respective fields. Real data can be placed anywhere in their field. A brief description of the variables used in the data file is given below. A number of these input parameters are illustrated graphically in Fig. A.6.

#### VARIABLE DESCRIPTION

VARIABLE	DESCRIPTION
<b>TITLE</b>	An 80 character title for the grid.
<b>NBLROW</b>	Number of blade rows. must be $\leq$ PARAMETER MROW.
<b>IGEOM</b>	Internal/external geometry parameter (not used).
<b>ITHETA</b>	Theta input format for blade definitions: if = 1, a mean camber line and tangential thickness are input, if = 2, surface theta coordinates are input (preferred).
<b>IWRITE</b>	Format of mesh (not used).
<b>NINPLS</b>	Inlet plane parameter (not used).
<b>NEXPLS</b>	Exit plane parameter (not used).
<b>NBLPTZ</b>	Number of points in the axial direction within the bounds of a blade row (see Fig. A.6). This must be an odd number. For a reasonably accurate solution, the number of points across the blade row should be

```

+TITLE-----
  GENERIC DUCTED PROPFAN-DUCTED CLIPPED SR7 GEOMETRY
+NBLROW--+IGEOM--+ITHETA--+IWRITE--+NCOWL-----+
      1      2      2      2
+NINPLS--+NEXPLS--+NBLPTZ--+NBLPTR--+NBLPTT--+--ISOLVE+--ITER-----+
      1      1      15     15     11      1      50
+ZINLET--+ZEXIT--+--DELTA--+ELFACZ--+ELFACR-----+
-0.50000  0.40000  0.10    0.2    0.11
+RATIN--+RATEX--+RATBB--+RATBLZ--+RATBLR--+RATBLT--+RATTOB-----+
  1.32000  1.32000  1.20000  1.20000  1.10000  1.05000  1.26000
*****
+Nsplit-----+
      4
*****
+NHUB-----+
      53
+ZHUB,RHUB+-----+
-80.00000 -20.00000 -0.45000 -0.40000 -0.37000 -0.35000 -0.33600 -0.33000
-0.32685 -0.32336 -0.31773 -0.31097 -0.30302 -0.29383 -0.28285 -0.25793
-0.22232 -0.18130 -0.13606 -0.09774 -0.07757 -0.07042 -0.06853 -0.06180
-0.05049 -0.03837 -0.02810 -0.01930 -0.01181 -0.00522  0.00080  0.00653
  0.01215  0.01808  0.02469  0.03237  0.04148  0.05238  0.06434  0.07312
  0.07788  0.07965  0.09454  0.11764  0.15665  0.21074  0.27769  0.35493
  0.43958  0.58799  0.75179  0.92019  1.09000
  0.00200  0.00200  0.00200  0.00200  0.00260  0.00400  0.00600  0.00800
  0.01000  0.01300  0.02177  0.03076  0.03877  0.04635  0.05436  0.06875
  0.08364  0.09408  0.09971  0.10164  0.10204  0.10239  0.10251  0.10301
  0.10421  0.10611  0.10819  0.11031  0.11257  0.11463  0.11653  0.11837
  0.12065  0.12307  0.12572  0.12874  0.13223  0.13630  0.14061  0.14368
  0.14530  0.14590  0.15081  0.15819  0.16767  0.17458  0.17754  0.17486
  0.16806  0.15661  0.15465  0.15465  0.15465
*****
+Nsplitnsnub -----+
      7
+Zspl,Rspl+-----+
  0.04000  0.03000 -0.03000 -0.04000 -0.01000  0.01000  0.04000
  0.34000  0.33500  0.33500  0.34000  0.34500  0.34500  0.34000
*****
+NSHR-----+
      75
+ZSHR,RSHR+-----+
  0.25000  0.24946  0.24786  0.24519  0.24148  0.23673  0.23097  0.22422
  0.21650  0.20786  0.19834  0.18796  0.17677 -0.16483 -0.17677 -0.18796
-0.19834 -0.20786 -0.21650 -0.22422 -0.23097 -0.23673 -0.24148 -0.24519
-0.24786 -0.24946 -0.25000 -0.24946 -0.24786 -0.24519 -0.24148 -0.23673
-0.23097 -0.22422 -0.21650 -0.20786 -0.19834 -0.18796 -0.17677 -0.16483
-0.15219 -0.13889 -0.12500 -0.11057 -0.09567 -0.08036 -0.06471 -0.04878
-0.03263 -0.01635  0.00000  0.01635  0.03263  0.04877  0.06471  0.08036
  0.09567  0.11057  0.12500  0.13889  0.15219  0.16483  0.17677  0.18796
  0.19834  0.20786  0.21650  0.22422  0.23097  0.23673  0.24148  0.24519
  0.24786  0.24946  0.25000
  0.41000  0.40990  0.40990  0.40990  0.40990  0.40990  0.40990  0.40990
  0.40990  0.40990  0.40986  0.40980  0.41473  0.40941  0.40436  0.40439
  0.40448  0.40464  0.40489  0.40525  0.40562  0.40608  0.40681  0.40744
  0.40813  0.40889  0.41000  0.41075  0.41166  0.41302  0.41468  0.41616
  0.41760  0.41917  0.42066  0.42209  0.42348  0.42480  0.42602  0.42715
  0.42818  0.42908  0.42985  0.43047  0.43092  0.43119  0.43125  0.43111
  0.43077  0.43025  0.42955  0.42870  0.42772  0.42666  0.42554  0.42438
  0.42321  0.42204  0.42089  0.41975  0.41866  0.41760  0.41659  0.41563
  0.41474  0.41390  0.41312  0.41240  0.41174  0.41118  0.41072  0.41038
  0.41016  0.41004  0.41000
*****
+NTIP-----+
      2
+ZTIP,RTIP+-----+
-10.00000  10.00000
  0.75000  0.75000

```

Figure A.5: Sample input data file for grid generation

```

*****
+NBLD-----+NBLCRC--+NPPRC--+IPCH--+ZPCA--+THPCA--+BETA--+ITIPC--+
      8      6      21      0      0.0000      0.0000      60.2000      0
+ZBLA,RBLA,THBLA,TTBLA--+-----+-----+-----+-----+-----+-----+-----+
-0.07042 -0.06259 -0.05477 -0.04699 -0.03924 -0.03153 -0.02386 -0.01628
-0.00878 -0.00130 0.00619 0.01346 0.02074 0.02804 0.03535 0.04269
0.05004 0.05742 0.06481 0.07222 0.07965 0.08708 0.09451 0.10194
-0.08927 -0.08096 -0.07266 -0.06436 -0.05608 -0.04782 -0.03959 -0.03140
-0.02326 -0.01519 -0.00719 0.00081 0.00875 0.01654 0.02436 0.03219
0.04005 0.04792 0.05581 0.06372 0.07164 0.07956 0.08748 0.09540
-0.10222 -0.09346 -0.08470 -0.07594 -0.06718 -0.05844 -0.04971 -0.04101
-0.03235 -0.02372 -0.01517 -0.00667 0.00183 0.01025 0.01855 0.02690
0.03526 0.04363 0.05203 0.06043 0.06886 0.07729 0.08571 0.09414
-0.11092 -0.10175 -0.09257 -0.08339 -0.07421 -0.06504 -0.05588 -0.04674
-0.03762 -0.02853 -0.01948 -0.01050 -0.00152 0.00741 0.01626 0.02510
0.03396 0.04283 0.05172 0.06062 0.06953 0.07844 0.08735 0.09626
-0.10567 -0.09655 -0.08743 -0.07831 -0.06919 -0.06008 -0.05097 -0.04189
-0.03281 -0.02376 -0.01475 -0.00577 0.00321 0.01214 0.02102 0.02992
0.03883 0.04775 0.05668 0.06562 0.07457 0.08351 0.09245 0.10139
-0.06358 -0.05613 -0.04869 -0.04125 -0.03381 -0.02638 -0.01895 -0.01153
-0.00411 0.00330 0.01070 0.01810 0.02549 0.03289 0.04029 0.04769
0.05509 0.06249 0.06990 0.07731 0.08472 0.09213 0.09954 0.10695
0.10239 0.10294 0.10369 0.10469 0.10595 0.10746 0.10915 0.11119
0.11351 0.11588 0.11825 0.12120 0.12414 0.12704 0.12989 0.13269
0.13544 0.13813 0.14078 0.14337 0.14590 0.14843 0.15096 0.15349
0.14224 0.14233 0.14263 0.14312 0.14379 0.14471 0.14589 0.14732
0.14897 0.15095 0.15319 0.15546 0.15789 0.16079 0.16361 0.16639
0.16910 0.17176 0.17438 0.17693 0.17943 0.18205 0.18468 0.18731
0.17874 0.17901 0.17911 0.17927 0.17967 0.18025 0.18105 0.18213
0.18345 0.18499 0.18689 0.18902 0.19118 0.19357 0.19638 0.19906
0.20169 0.20427 0.20679 0.20926 0.21167 0.21408 0.21649 0.21890
0.22445 0.22475 0.22492 0.22504 0.22525 0.22562 0.22620 0.22701
0.22805 0.22934 0.23084 0.23273 0.23467 0.23678 0.23921 0.24169
0.24409 0.24644 0.24874 0.25098 0.25317 0.25536 0.25755 0.25974
0.27612 0.27636 0.27645 0.27653 0.27679 0.27718 0.27773 0.27849
0.27945 0.28059 0.28199 0.28357 0.28518 0.28704 0.28914 0.29112
0.29306 0.29497 0.29683 0.29863 0.30040 0.30217 0.30394 0.30571
0.40500 0.40516 0.40532 0.40548 0.40564 0.40580 0.40596 0.40612
0.40628 0.40644 0.40660 0.40676 0.40692 0.40708 0.40724 0.40740
0.40756 0.40772 0.40788 0.40804 0.40820 0.40836 0.40852 0.40868
-0.22623 -0.11448 -0.06580 -0.00593 0.03783 0.07747 0.11208 0.13846
0.15770 0.17116 0.17992 0.18148 0.18524 0.18762 0.18808 0.18672
0.18416 0.18144 0.17914 0.17788 0.17801 0.17814 0.17827 0.17840
-0.21743 -0.16069 -0.13115 -0.09805 -0.07011 -0.04451 -0.02039 0.00151
0.02138 0.03888 0.05401 0.06720 0.07870 0.08821 0.09698 0.10506
0.11278 0.12067 0.12867 0.13656 0.14430 0.15204 0.15978 0.16752
-0.22009 -0.17785 -0.15193 -0.12440 -0.09990 -0.07691 -0.05491 -0.03430
-0.01496 0.00299 0.01943 0.03450 0.04855 0.06161 0.07367 0.08531
0.09651 0.10748 0.11820 0.12866 0.13889 0.14805 0.15620 0.16435
-0.22349 -0.18910 -0.16466 -0.13966 -0.11662 -0.09455 -0.07318 -0.05272
-0.03311 -0.01442 0.00338 0.02025 0.03646 0.05204 0.06701 0.08151
0.09562 0.10941 0.12280 0.13576 0.14805 0.15978 0.17049 0.18020
-0.19509 -0.16591 -0.14364 -0.12121 -0.09998 -0.07939 -0.05926 -0.03966
-0.02054 -0.00199 0.01598 0.03346 0.05037 0.06683 0.08275 0.09822
0.11322 0.12781 0.14191 0.15547 0.16812 0.18077 0.19242 0.20407
-0.09282 -0.07252 -0.05588 -0.03926 -0.02329 -0.00766 0.00771 0.02279
0.03766 0.05229 0.06671 0.08089 0.09482 0.10855 0.12203 0.13526
0.14816 0.16076 0.17300 0.18478 0.19570 0.20662 0.21754 0.22846
-0.22623 -0.25766 -0.25881 -0.25715 -0.25046 -0.24091 -0.22859 -0.21185
-0.19041 -0.16474 -0.13510 -0.10277 -0.07418 -0.04316 -0.01028 0.02425
0.05918 0.09269 0.12376 0.15196 0.17801 0.20204 0.22408 0.24412
-0.21743 -0.22212 -0.20847 -0.19620 -0.18074 -0.16395 -0.14652 -0.12818
-0.10906 -0.08898 -0.06796 -0.04625 -0.02392 -0.00143 0.02083 0.04278
0.06427 0.08517 0.10544 0.12510 0.14430 0.16297 0.18112 0.19877
-0.22009 -0.21557 -0.19920 -0.18427 -0.16738 -0.14978 -0.13197 -0.11370
-0.09505 -0.07598 -0.05637 -0.03634 -0.01600 0.00444 0.02471 0.04480
0.06456 0.08392 0.10277 0.12103 0.13890 0.15627 0.17264 0.18801
-0.22349 -0.21227 -0.19376 -0.17620 -0.15761 -0.13876 -0.11988 -0.10086
-0.08176 -0.06255 -0.04321 -0.02382 -0.00435 0.01505 0.03434 0.05353
0.07264 0.09166 0.11059 0.12936 0.14805 0.16627 0.18394 0.20107
-0.19509 -0.18096 -0.16248 -0.14470 -0.12627 -0.10775 -0.08929 -0.07077
-0.05226 -0.03375 -0.01518 0.00331 0.02187 0.04033 0.05875 0.07715
0.09554 0.11384 0.13209 0.15024 0.16813 0.18577 0.20296 0.21961
-0.09282 -0.07935 -0.06428 -0.04962 -0.03484 -0.02008 -0.00543 0.00917
0.02374 0.03826 0.05271 0.06713 0.08149 0.09586 0.11021 0.12457
0.13894 0.15326 0.16758 0.18182 0.19570

```

19 or larger.

**NBLPTR** Number of points in the radial direction within the bounds of a blade row (see Fig. A.6). This must be an odd number. For a reasonably accurate solution, the number of points across the blade span should be 17 or larger.

**NBLPTT** Number of points between airfoils in the circumferential direction. This must be an odd number  $\leq$  PARAMETER MTP.

**ISOLVE** Elliptic smoothing procedure control parameter:

if = 0, no smoothing is done.

if = 1, elliptic smoothing is done with some preservation of axial and radial clustering.

**ITER** Number of iterations for the elliptic smoothing procedure (100 recommended).

**ZINLET** Axial location of the inlet boundary plane.

**ZEXIT** Axial location of the exit boundary plane.

**DELF** Distance from solid bodies (blade row, hub, splitter, etc.) over which grid spacing is preserved during elliptic smoothing.

**ELFACZ** Strength factor for the elliptic smoothing operator in the axial direction (0.2 recommended).

**ELFACR** Strength factor for the elliptic smoothing operator in the radial direction (0.2 recommended).

The elliptic smoothing factors have a significant effect on the operation and its stability and the effects are grid dependent. A set of factors which produce good

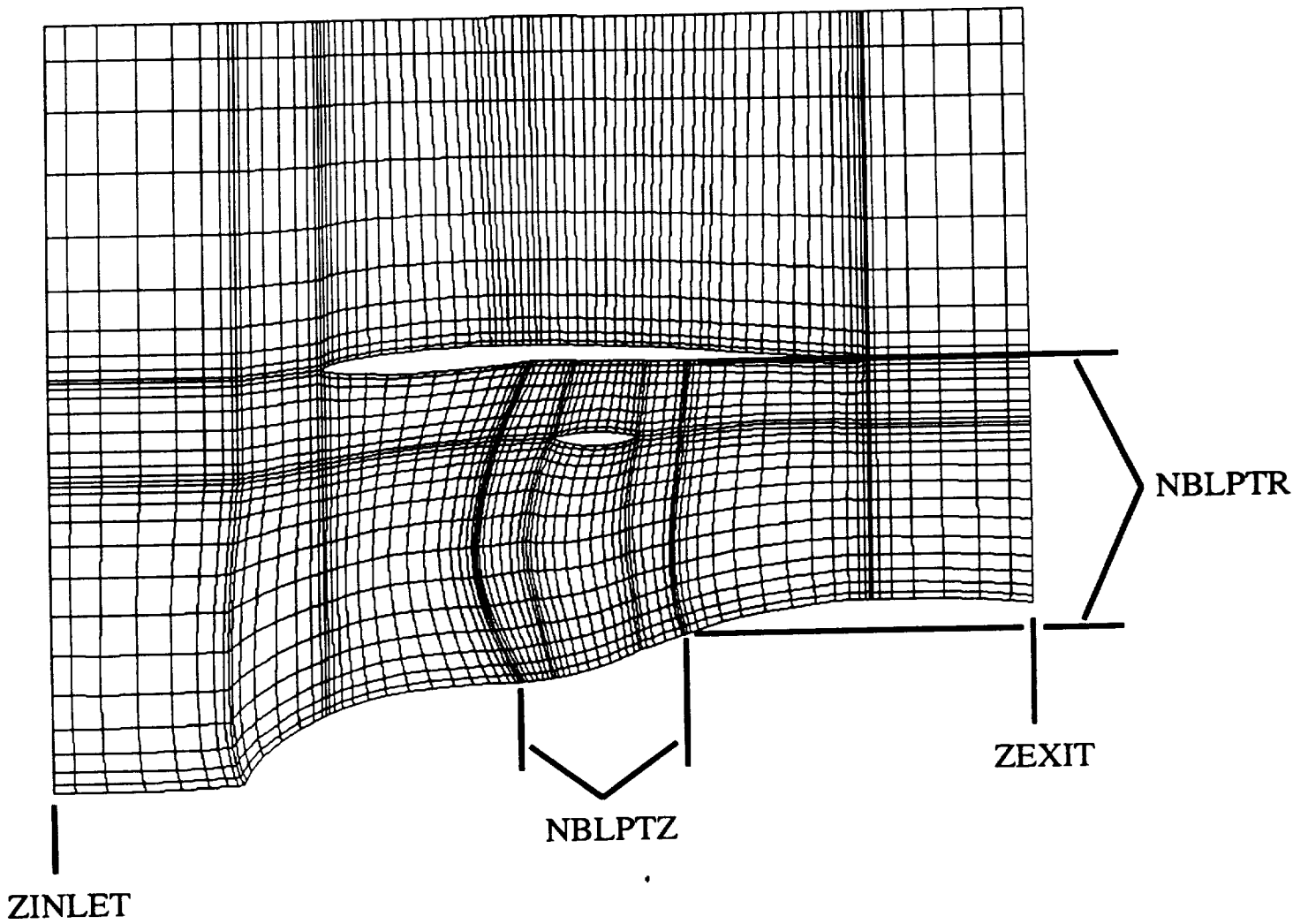


Figure A.6: Grid from sample input data illustrating grid parameters

smoothing for one grid may cause the elliptic smoothing procedure to fail on a different grid. If such an instability occurs the user should reduce the strength factors and run again.

The next 7 variables in the input file (**RATIN RATEX RATBB RATBLZ RATBLR RATBLT RATTOB**) are not used to generate the grid. The space for these variables is in the input file for potential future use and to maintain commonality between this file and other grid generator input files.

The following section of the input file defines the radial boundaries (e.g., the hub flow path and the outer boundary of the grid domain) and the radial flow splitters (e.g., part span shroud, cowl, core/bypass split). The boundary data should be ordered according to the user's desired radial blocking arrangement from the center line of the geometry outward. The hub defining section is first, followed by any radial splitters, and the outer boundary definition is last. There must always be a hub and outer boundary section and the total number of radial boundaries must be  $\leq$  **PARAMETER MROW**.

These boundary data consist of pairs (z,r) of coordinates ordered in a way particular to the boundary they define. The hub and outer boundary data are ordered from the most forward position to the aft most data point. The other radial splitters are ordered beginning with the most aft data point followed by the other points in a clock-wise direction ending with the most aft data point again.

**NSPLIT** Total number of radial splitter and boundary defining sections (hub, cowl, outer boundary, flow splitters, etc.).

The next line in the input file has a special function. Ordinarily, the line just



serves to separate the splitter defining data section by a line of asterisks:

```
***** . . .
```

with the data section following. However if the first character in the line is the “@” symbol, the program is directed to read the splitter defining data section from a separate file indicated on the current line. For instance, if the line on the original input file is as follows:

```
@hub1.d ***** . . .
```

the file *hub1.d* will be read for that particular radial defining section. The defining data section for any or all radial boundaries and blade rows can be input this way. A sample input file containing this redirected input option is seen in Fig. A.7. Figure A.8 shows an example of a defining data section file.

**NPSPL** Number of (z,r) radial boundary coordinate pairs to be read.

**ZSPLT,RSPLT** Radial splitter or boundary coordinates. All of the axial coordinates are given first (8 per line), followed separately by the radial coordinates.

The number of points input is determined by **NPSPL**.

The next section of the input file describes the blade rows themselves. The blades are defined on slices through the airfoil with identical axial and radial coordinates on each surface of the airfoil. The radial cross section description is illustrated in Fig. A.9. The airfoil coordinate description along each cross section is illustrated in Fig. A.10. The airfoil slices are given in order from the hub to the tip. The hub and

```

+TITLE-----
  GENERIC DUCTED PROPFAN-DUCTED CLIPPED SR7 GEOMETRY
+NBLROW--+IGEOM--+ITHETA--+IWRITE--+NCOWL--+-----+
      1          2          2          2
+NINPLS--+NEXPLS--+NBLPTZ--+NBLPTR--+NBLPTT--+--ISOLVE--+--ITER--+-----+
      1          1          15         15         11         1          50
+ZINLET--+ZEXIT--+--DELFF--+--ELFACZ--+--ELFACR--+-----+
      -0.50000   0.40000   0.10      0.2      0.11
+RATIN--+RATEX--+--RATBB--+--RATBLZ--+--RATBLR--+--RATBLT--+--RATTOB--+-----+
      1.32000   1.32000   1.20000   1.20000   1.10000   1.05000   1.26000
*****
+Nsplit-----+
      4
@hub1.dat*****
@snub1.dat *****
@shroud1.dat *****
*****
+NTIP--+-----+
      2
+ZTIP,RTIP+-----+
      -10.00000   10.00000
        0.75000   0.75000
@blad1.d*****

```

Figure A.7: Sample input data file for exhibiting redirected input format

```

+NHUB----- hub1.dat -----+
      53
+ZHUB,RHUB+-----+
-80.00000 -20.00000 -0.45000 -0.40000 -0.37000 -0.35000 -0.33600 -0.33000
-0.32685 -0.32336 -0.31773 -0.31097 -0.30302 -0.29383 -0.28285 -0.25793
-0.22232 -0.18130 -0.13606 -0.09774 -0.07757 -0.07042 -0.06853 -0.06180
-0.05049 -0.03837 -0.02810 -0.01930 -0.01181 -0.00522 0.00080 0.00653
 0.01215 0.01808 0.02469 0.03237 0.04148 0.05238 0.06434 0.07312
 0.07788 0.07965 0.09454 0.11764 0.15665 0.21074 0.27769 0.35493
 0.43958 0.58799 0.75179 0.92019 1.09000
 0.00200 0.00200 0.00200 0.00200 0.00260 0.00400 0.00600 0.00800
 0.01000 0.01300 0.02177 0.03076 0.03877 0.04635 0.05436 0.06875
 0.08364 0.09408 0.09971 0.10164 0.10204 0.10239 0.10251 0.10301
 0.10421 0.10611 0.10819 0.11031 0.11257 0.11463 0.11653 0.11837
 0.12065 0.12307 0.12572 0.12874 0.13223 0.13630 0.14061 0.14368
 0.14530 0.14590 0.15081 0.15819 0.16767 0.17458 0.17754 0.17486
 0.16806 0.15661 0.15465 0.15465 0.15465

```

Figure A.8: Sample of a defining data section file (hub1.dat) for redirected input

tip cross section may extend beyond the defining flow path boundaries but if a tip clearance is required, the last airfoil slice must define the actual tip of the blade.

Each data section defining a blade row is separated by a line of asterisks (see Fig. A.5) as was done for the radial boundary defining sections. And as was possible with the radial boundary data, the blade row input can be redirected to a particular file if the "@" symbol is in that line's first column (see Fig. A.7).

The following variables define the blade row.

**NBLD** Number of blades in the blade row.

**NBLRCS** Number of radial cross sections used to define the blade (see Fig. A.9).

**NPPRC** Number of axial points per radial cross section used to define the blade (see Fig. A.10).

An option is provided in *TIGG3D* to arbitrarily adjust the  $3/4$  radius blade setting angle without modifying the blade geometry parameters. This assumes that the pitch change axis is perpendicular to the axis of rotation, and only a solid body rotation can occur (no deflections). This procedure can occasionally produce unexpected results if the blade geometry is not defined beyond the confines of the hub surface, especially if the hub is highly ramped. Also blade tips may interfere with shroud boundaries and produce peculiar results if incorrect blade setting angles or blade tips are input.

**IPCH** Pitch change trigger: if = 0 no pitch change is performed, if = 1 the pitch change variables **ZPCA**, **THPCA**, and **BETA34** are read in and the blade setting angle is adjusted to the value specified by **BETA34** (see Fig. A.11).

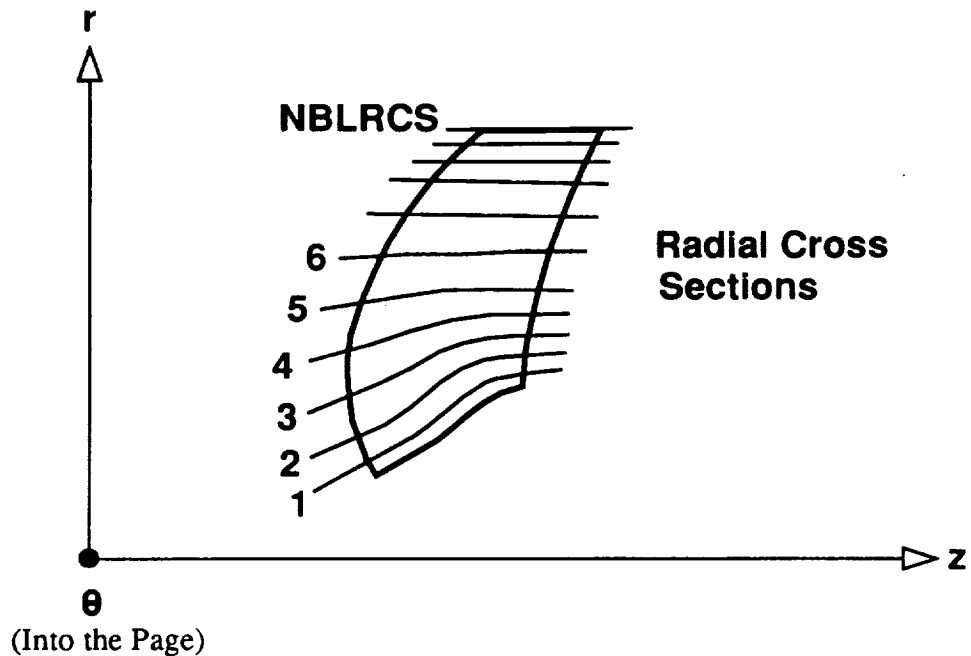


Figure A.9: Radial cross section of blade geometry

**ZPCA** Axial location of the blade pitch change axis. The pitch change axis is assumed perpendicular to the axis of rotation (see Fig. A.11).

**THPCA** Circumferential position of the blade pitch change axis. The pitch change axis is assumed perpendicular to the axis of rotation.

**BETA34** Absolute 3/4 radius blade setting angle. This angle is measured positive from the plane of rotation. The blade setting angle reference plane is the same for either positive or negative rotation (see Fig. A.11).

**ITIPC** Tip clearance trigger: if = 0 no tip clearance, if = 1 tip clearance is desired with the blade tip defined by the last airfoil section.

**ZBLA,RBLA,** Axial and radial blade coordinates.

**THBLA,TTBLA** Circumferential blade coordinates.

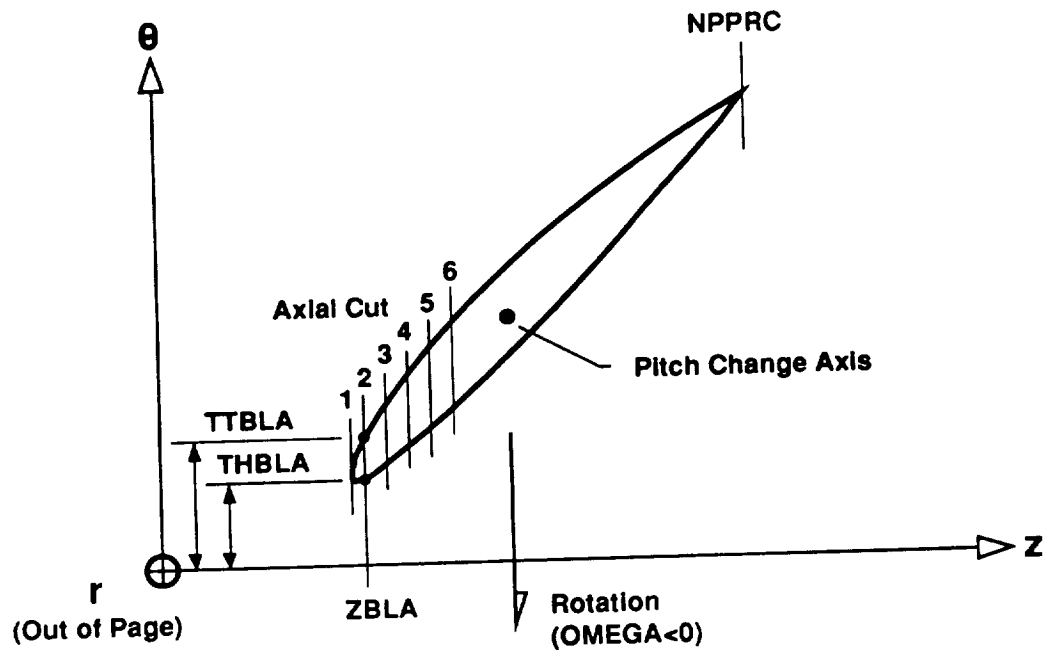


Figure A.10: Blade defining coordinates

Each of the blade axial coordinates is given first (8 per line) running chordwise from leading edge to trailing edge for each cross section, (cross sections are given from hub to tip) followed by all the radial coordinates. The blade circumferential coordinates are given next, and may be in two different formats depending on the variable **ITHETA**. The preferred format is **ITHETA=2**, where the actual tangential surface coordinates are given ordered by increasing theta. For a blade row rotating in the negative direction, the surfaces would be ordered as pressure then suction surface (see Fig. A.10).

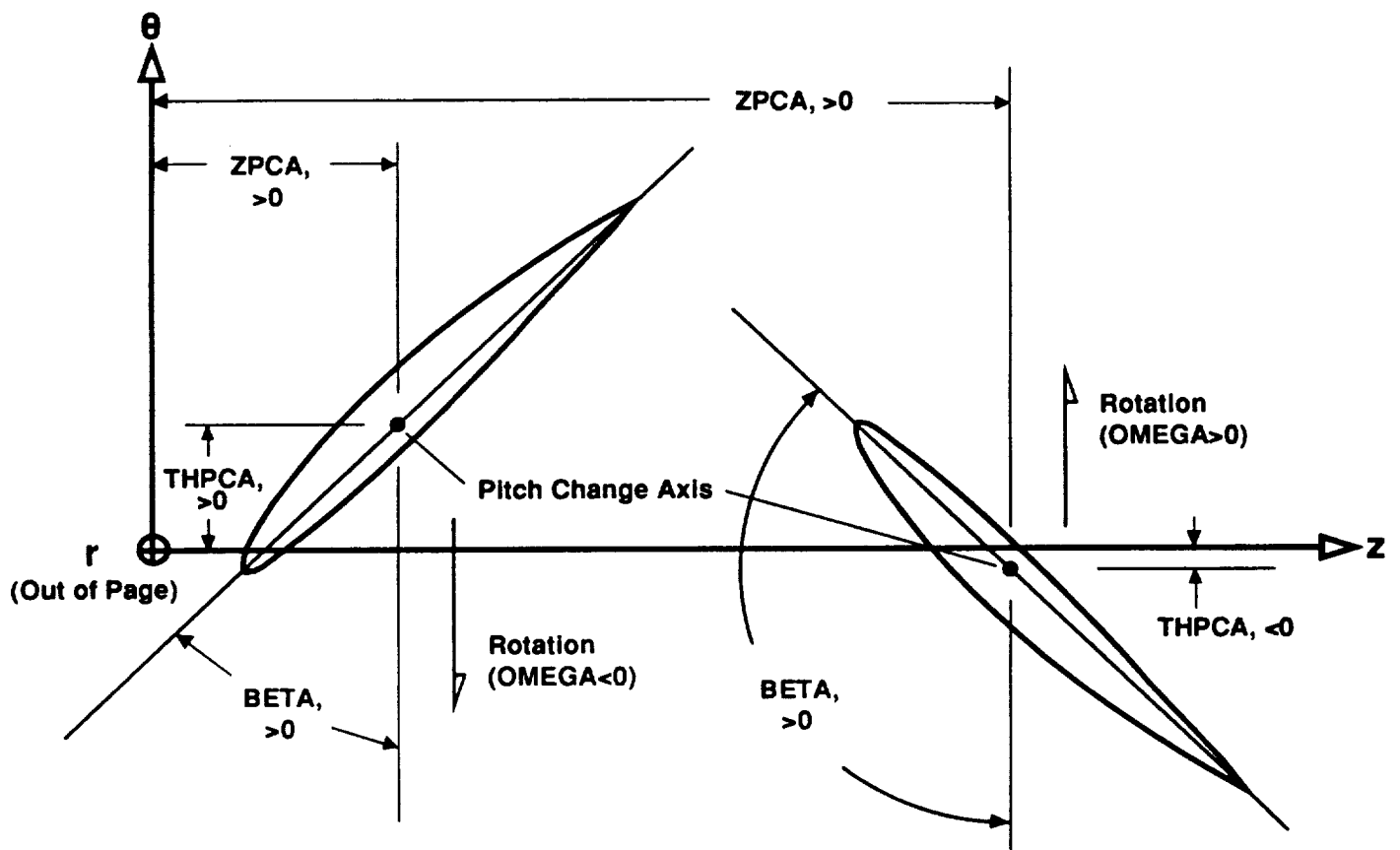


Figure A.11: Blade setting angle definition

#### A.4 Input File Description: Block Boundary Input Section

The previous section dealt with the basic minimum geometry input needed to form a grid. If the input file is made up of that initial input alone, procedures in the code take the input and generate grid block boundaries (flowpath and station data) from it. This block boundary data is often manipulated by the user in the graphic-interactive version of the code to produce a better quality grid. This block boundary and grid distribution data can be written out and appended to the input file. This section describes that portion of the input file.

An example of this input data is seen in Fig. A.12. This input section follows the blade geometry section described above. A graphic arrangement of the block data described in this section is seen in Fig. A.13.

After a demarcation line five lines of title text are read. Following that are the number of grid block rows and the total number of blocks. This data is read in free format.

The next section defines the flow path boundaries of all the grid blocks. The flow path boundaries for a typical block are seen in Fig. 3.3. The "shroud" section is the upper most boundary and the "hub" section is the lower boundary for every block. The input file has the "shroud" data for all of the blocks, followed by a section containing the "hub" data for all the blocks. For each flowpath section the number of points defining that section (NPTS) are read in followed by the paired (Z,R) coordinates. All the data is in free format.

Data describing the grid block stations is next in the input file. Each block has two stations as seen in Fig. 3.3. Blocks that are adjacent to each other on a station

boundary are identically defined on that common boundary but each block has a separate station. Thus the points describing station two are the same for station three, and the points for station four are the same as station five, and so forth. This relationship is displayed in Fig. 3.3 . For each station, the number of points defining that section (NST) are read in a formatted line followed by the paired (Z,R) coordinates in a free format.

The actual grid parameters for the blocks that make up the meridional section are contained in the next input section. The axial grid point parameters are listed for every block in the grid and then the radial parameters. The data is read in a free format. A reference list for the parameters and what they control is given below.

**NBKP** Number of axial points in the individual block.

**NB** Number of blades in the grid block.

**SLE** Axial spacing parameter for the forward portion of the grid block (axial length ratio of the first two cells).

**STE** Axial spacing parameter for the aft portion of the grid block (axial length ratio of the last two cells).

**DLE** Axial spacing parameter for the forward portion of the grid block (physical distance of the first axial cell length).

**DTE** Axial spacing parameter for the aft portion of the grid block (physical distance of the last axial cell length).

**NSP** Number of radial points in the individual block.

**SRH** Radial spacing parameter for the lower portion of the grid block (spanwise length ratio of the first two cells).



```

+-----begin-----+
GENERIC DUCTED PROPFAN-DUCTED CLIPPED SR7 GEOMETRY

*****number of rows*****
3
*****number of blocks*****
24
*****Flowpath input coordinates*****
+shroud section+---npts+---+
1 2
+-----Z-----+-----R-----+
-0.5000000 0.3052801
-0.3268500 0.3052801
+shroud section+---npts+---+
2 2
+-----Z-----+-----R-----+
-0.3268500 0.3052801
-0.2504528 0.3177684
+shroud section+---npts+---+
3 3
+-----Z-----+-----R-----+
-0.2504528 0.3177684
-0.1730233 0.3301041
-8.9789949E-02 0.3377806

.
.
.

+shroud section+---npts+---+
24 2
+-----Z-----+-----R-----+
0.2500000 0.7500000
0.4000000 0.7500000
+hub section+---npts+---+
1 3
+-----Z-----+-----R-----+
-0.5000000 2.0000001E-03
-0.3500000 4.0000002E-03
-0.3262142 5.8356663E-03
+hub section+---npts+---+
2 6
+-----Z-----+-----R-----+
-0.3262142 5.8356663E-03
-0.3205237 1.8243141E-02
-0.3109700 3.0760000E-02
-0.2938300 4.6349999E-02
-0.2735793 6.0536869E-02
-0.2500000 7.2566383E-02

.
.
.

+hub section+---npts+---+
24 2
+-----Z-----+-----R-----+
0.2500000 0.4100000
0.4000000 0.4100000
*****station input data*****
+---station+---+id+---+idtype+---+nst+---+
1.0000000 2
+-----Z-----+-----R-----+
-0.5000000 2.0000001E-03
-0.5000000 0.3052801
+---station+---+id+---+idtype+---+nst+---+
2.0000000 4
+-----Z-----+-----R-----+
-0.3262142 5.8356663E-03
-0.3329690 2.6006460E-02
-0.3384379 9.4914436E-02
-0.3268500 0.3052801
+---station+---+id+---+idtype+---+nst+---+
3.0000000 4
+-----Z-----+-----R-----+
-0.3262142 5.8356663E-03
-0.3329690 2.6006460E-02
-0.3384379 9.4914436E-02
-0.3268500 0.3052801

```

Figure A.12: Sample block boundary input data section

```

+---station---+---id---+---idtype---+---nst---+
48.0000000      2
+-----Z-----+-----R-----+
0.4000000      0.4100000
0.4000000      0.7500000
*****general information*****
+---nbkp---+---nb---+---sle---+---ste---+---dle---+---dte---+
9 0 0.100E+01 0.100E+01 0.173E-01 0.400E-02
9 0 0.100E+01 0.100E+01 0.400E-02 0.300E-02
17 0 0.100E+01 0.100E+01 0.300E-02 0.200E-02
9 8 0.100E+01 0.100E+01 0.200E-02 0.237E-02
13 8 0.100E+01 0.100E+01 0.300E-02 0.300E-02
7 8 0.100E+01 0.100E+01 0.300E-02 0.200E-02
15 0 0.100E+01 0.100E+01 0.200E-02 0.300E-02
9 0 0.100E+01 0.100E+01 0.300E-02 0.150E-01
9 0 0.100E+01 0.100E+01 0.173E-01 0.400E-02
9 0 0.100E+01 0.100E+01 0.400E-02 0.300E-02
17 0 0.100E+01 0.100E+01 0.300E-02 0.200E-02
9 8 0.100E+01 0.100E+01 0.200E-02 0.237E-02
13 8 0.100E+01 0.100E+01 0.300E-02 0.300E-02
7 8 0.100E+01 0.100E+01 0.300E-02 0.200E-02
15 0 0.100E+01 0.100E+01 0.200E-02 0.300E-02
9 0 0.100E+01 0.100E+01 0.300E-02 0.150E-01
9 0 0.100E+01 0.100E+01 0.173E-01 0.400E-02
9 0 0.100E+01 0.100E+01 0.400E-02 0.300E-02
17 0 0.100E+01 0.100E+01 0.300E-02 0.895E-02
9 0 0.100E+01 0.100E+01 0.579E-02 0.579E-02
13 0 0.100E+01 0.100E+01 0.835E-02 0.835E-02
7 0 0.100E+01 0.100E+01 0.894E-02 0.894E-02
15 0 0.100E+01 0.100E+01 0.961E-02 0.300E-02
9 0 0.100E+01 0.100E+01 0.300E-02 0.150E-01
+---nsp---+---srh---+---srs---+---dxh---+---dxs---+
15 0.100E+01 0.100E+01 0.700E-02 0.400E-02
15 0.100E+01 0.100E+01 0.700E-02 0.400E-02
15 0.100E+01 0.100E+01 0.700E-02 0.400E-02
15 0.100E+01 0.100E+01 0.700E-02 0.400E-02
15 0.100E+01 0.100E+01 0.700E-02 0.400E-02
15 0.100E+01 0.100E+01 0.700E-02 0.400E-02
15 0.100E+01 0.100E+01 0.700E-02 0.400E-02
9 0.100E+01 0.100E+01 0.403E-02 0.403E-02
9 0.100E+01 0.100E+01 0.403E-02 0.403E-02
9 0.100E+01 0.100E+01 0.403E-02 0.403E-02
9 0.100E+01 0.100E+01 0.403E-02 0.403E-02
9 0.100E+01 0.100E+01 0.403E-02 0.403E-02
9 0.100E+01 0.100E+01 0.403E-02 0.403E-02
9 0.100E+01 0.100E+01 0.403E-02 0.403E-02
9 0.100E+01 0.100E+01 0.403E-02 0.403E-02
11 0.100E+01 0.100E+01 0.403E-02 0.319E-01
11 0.100E+01 0.100E+01 0.403E-02 0.319E-01
11 0.100E+01 0.100E+01 0.403E-02 0.319E-01
11 0.100E+01 0.100E+01 0.403E-02 0.319E-01
11 0.100E+01 0.100E+01 0.403E-02 0.319E-01
11 0.100E+01 0.100E+01 0.403E-02 0.319E-01
11 0.100E+01 0.100E+01 0.403E-02 0.319E-01
+-----end of flowpath-----+
+-----Tangential Parameters-----+
+--- for blade row 1 nbzsl: 1 nbzste: 1
+---block---+---st1---+---st2---+---dt1---+---dt2---+---nblptt---+
1 0.100E+01 0.100E+01 0.403E-02 0.000E+00 11
2 0.100E+01 0.100E+01 0.403E-02 0.000E+00
3 0.100E+01 0.100E+01 0.403E-02 0.000E+00
4 0.100E+01 0.100E+01 0.403E-02 0.000E+00
5 0.100E+01 0.100E+01 0.403E-02 0.000E+00
6 0.100E+01 0.100E+01 0.403E-02 0.000E+00
7 0.100E+01 0.100E+01 0.403E-02 0.000E+00
8 0.100E+01 0.100E+01 0.403E-02 0.000E+00
9 0.100E+01 0.100E+01 0.403E-02 0.000E+00
10 0.100E+01 0.100E+01 0.403E-02 0.000E+00
11 0.100E+01 0.100E+01 0.403E-02 0.000E+00
12 0.100E+01 0.100E+01 0.403E-02 0.000E+00
13 0.100E+01 0.100E+01 0.403E-02 0.000E+00
14 0.100E+01 0.100E+01 0.403E-02 0.000E+00
15 0.100E+01 0.100E+01 0.403E-02 0.000E+00
16 0.100E+01 0.100E+01 0.403E-02 0.000E+00
17 0.100E+01 0.100E+01 0.403E-02 0.000E+00
18 0.100E+01 0.100E+01 0.403E-02 0.000E+00
19 0.100E+01 0.100E+01 0.403E-02 0.000E+00
20 0.100E+01 0.100E+01 0.403E-02 0.000E+00
21 0.100E+01 0.100E+01 0.403E-02 0.000E+00
22 0.100E+01 0.100E+01 0.403E-02 0.000E+00
23 0.100E+01 0.100E+01 0.403E-02 0.000E+00
24 0.100E+01 0.100E+01 0.403E-02 0.000E+00

```

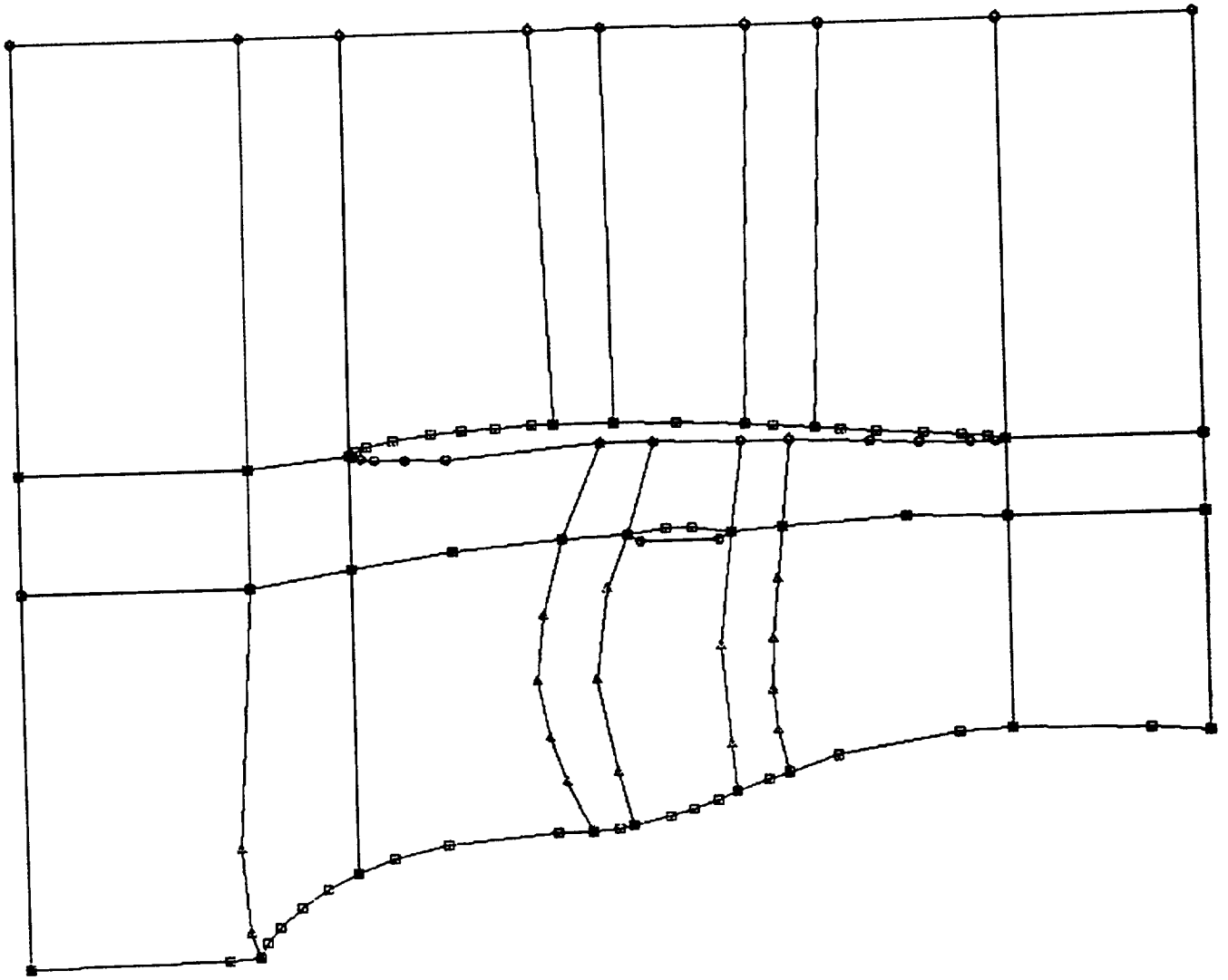


Figure A.13: Block boundary input data

**SRS** Radial spacing parameter for the upper portion of the grid block (spanwise length ratio of the last two cells).

**DXH** Radial spacing parameter for the lower portion of the grid block (physical distance of the first radial cell length).

**DXS** Radial spacing parameter for the upper portion of the grid block (physical distance of the first radial cell length).

The last section of the input file is made up of the tangential grid parameters. The first line containing the variables **NBZSLE** and **NBZSTE** is read in a formatted statement. The other data are read in a free format and a description of the parameters is given below. If there is more than one blade row, the parameters for that row follow the first, starting with the formatted data line.

**NBZSLE** The number of blocks forward of the leading edge of the blade row through which the tangential grid angle goes from the leading edge value to zero slope.

**NBZSTE** The number of blocks aft of the trailing edge of the blade row through which the tangential grid angle goes from the trailing edge value to zero slope.

**ST1** Currently unused spacing parameter.

**ST2** Currently unused spacing parameter.

**DT1** Tangential spacing parameter for the grid block (physical distance of the first tangential cell length).

**DT2** Currently unused spacing parameter.

**NBLPTT** The number of grid points in the tangential direction. This value

overrides the one given in the first input section.

## A.5 Output File Description

There are two basic output files types that **TIGG3D** generates. The first is an echo of the basic input section described earlier. The program assigns a name to this file by adding a “.new” suffix to the input file name. For example, if the input file has the name

*case.input*

then the echoed basic input section will be written to a file named

*case.input.new*

If that file already exists, then the user will be prompted to enter a different name for this output file.

Grid block data and tangential mesh parameters are also written to this file if the menu item **Write Output File** is selected in the main menu. This is usually done at the end of a graphic-interactive session so that any grid manipulations can be saved. The output file generated in this way can then function as a complete input file to generate a grid in the “batch” version of the code, or to regenerate a grid for continued manipulation in the graphic-interactive version.

The other output file is the full three-dimensional grid. The grid is in a machine independent format, and is compatible with the PLOT3D multiple-grid, binary file description. This output file for a single blade row grid is called *grid1.bin*. If there are multiple blade rows, then each three-dimensional grid produced follows the naming convention shown below.

## FILE DESCRIPTION

---

*grid1.bin* Three-dimensional grid based upon blade row number one.

*grid2.bin* Three-dimensional grid based upon blade row number two.

*grid3.bin* Three-dimensional grid based upon blade row number three.

*grid4.bin* Three-dimensional grid based upon blade row number four.

There is one last output file written and it contains messages from various routines of the program. The messages are rather cryptic and of little use to the user except for diagnostic purposes if the code is not functioning as expected. The file is called *fort.16*.

## A.6 Subroutine Description

A list of the grid generation program subroutines and their functions is given below for reference.

### SUBROUTINE DESCRIPTION

---

<b>ADDPNT</b>	Routine adds a point to either a block boundary or a grid.
<b>ADDSEG</b>	Routine to add a segment to a block flowpath boundary.
<b>ADDSTA</b>	Routine to add a station boundary to the block configuration.
<b>ARBETA</b>	Routine to compute tangential coordinates from a beta angle distribution.
<b>AXIS</b>	Routine to plot grid coordinate axis into the graphics window.
<b>BACKDR</b>	Routine sets up the logo for the program start up.
<b>BETAA</b>	Routine calculates beta angles from an axial, radial, and tangential set of coordinates.
<b>BEZCUR</b>	Routine adds points to a grid or block boundary along a bezier curve. Routine calls <b>SECCAL</b> and <b>BESPL</b> .
<b>BLKGR</b>	Routine to generate the meridional grid.
<b>BLOCKA</b>	Graphic routine for changing points and spacing for grid blocks in the axial direction.
<b>BLOCKM</b>	Block grid menu routine.
<b>BUILDG</b>	Routine to build a composite meridional grid from the individual blocks for the elliptic smoothing routine.
<b>CFCOE</b>	Routine to generate cubic spline coefficients.



- CFOUT** Cubic spline interpolation routine.
- CONVAS** Routine to convert array storage - used in conjunction with the SD-BLIB output routines.
- CURSP0** Mouse control routine for menu selection.
- DCSPL** Cubic spline routine. Calls **XDSPL** for coefficients and **XDINTP** for interpolation.
- DELPNT** Subroutine to delete a point from a grid or block boundary.
- DELSTA** Subroutine to delete a station block boundary.
- DMNMX** Minimum or maximum value finding routine.
- DOPQ** Routine to generate elliptic smoothing factors for the meridional grid.
- DSVAL** Routine to generate length distribution.
- EDITTL** Edit tool menu routine. Routine calls **EDTMEN** and **EMENU**.
- FLOWPM** Flow path menu routine. Routine calls **FL0MEN** and **PMENU**.
- GBMAKE** Routine to generate block boundary definitions for elliptic smoother.
- GENBLK** Routine to generate block boundaries from input data and radial bounds determined in **WORK2**.
- HPACK2** Routine to generate a hyperbolic tangent spacing distribution.
- INITAR** Routine to reinitialize arrays for a running restart of the program.
- INTSCC** Routine to find the intersection of two lines.
- ISSTA** Routine to initialize graphics window.
- ILOGO** Initial logo generation routine.
- MENTRY** Routine to manually input a point to a grid block.
- MOVEPT** Routine to move a grid point or a block boundary point.

- PL3D** Routine to condense grid blocks and generate 3-dimensional grid.
- PLOT2D** Routine to plot grid and block boundary data. Routine calls **FPPLT** for flowpaths, **STAPLT** for stations, and **BLKPLT** for grid blocks.
- PLOTTH** Plotting routine for a radial by tangential portion of the grid.
- POISSN** Poisson solver for the elliptic smoothing operation.
- READMU** Routine to read in the grid input file.
- ROTATE** Routine to adjust blade coordinates to the input blade set angle.
- SAVE** Routine to write set up parameters to the file *setup*.
- SCALEF** Routine to rescale grid parameters.
- SLICB** Routine to interpolate grid tangential locations from the airfoil input data.
- SMOOF** Routine to compute fast fourier transforms for elliptic smoothing functions. Routine calls **REALFT** and **FOUR1**.
- SMOOTH** Elliptic smoothing control routine.
- SPANCH** Graphic routine for changing points and spacing for grid blocks in the radial direction.
- SPLINE** Routine to insert a point into block boundary.
- SPROG** Exit menu control routine. Routine calls **QUITMN** and **QMENU**.
- STATM** Station menu control routine. Routine calls **STAMEN** and **SMENU**.
- SYMBOL** Routine to plot symbols in the graphics window.
- TABPTS** Routine to distribute points on a line. Routine calls **DSVAL**.
- TFINTRP** Transfinite interpolation routine for grid blocks.
- THCHAN** Graphic routine for changing points and spacing for the grid in the

tangential direction.

**THETAM** Tangential menu control routine.

**THFIND** Routine to generate tangential coordinates for the meridional grid blocks.

**THGRID** Routine to generate the tangential grid point distribution across the blade pitch.

**TMENU** Root menu control routine.

**WORK2** Routine to initialize the radial block bounds from input data.

**WRITEG** Routine to write the block boundary data to the output file.

**XTRACT** Routine to decompose elliptic smoothing composite grid into individual grid blocks.

**ZOOM** Routine to translate and magnify plot in the graphics window.

**ZOOMT** Routine to translate and magnify plot of the tangential grid in the graphics window.

In addition to fortran language subroutines, there are a few routines in the C programming language that are usually associated with the interactive graphics. A list of the files contained the routines is shown below.

#### **FILE DESCRIPTION**

---

**ASK.C** Routine to generate a button response question in the graphics window.

**BAR.C** A set of routines with graphic interaction of functions using slide bars.

**BUTTON.C** Graphic routines controlling many grid point and block boundary manipulations.

It should also be mentioned that a number of routines from the SDBLIB library are also included in the source code distribution, but are not defined here.

## A.7 *TIGG3D*: GRAPHIC-INTERACTIVE OPERATION

### A.7.1 Introduction

This section outlines the operations of the graphic-interactive version of the **TIGG3D** grid generator. Most of the operations covered are part of the original **TIGGERC** code and for further reference and instruction the user is referred to [1].

The grid generator is highly menu driven and organized and likewise this section is separated according to menus and options within those menus.

### A.7.2 Main Menu Options

Once the program has begun execution with the user entering the command **tigg3d** on the Silicon Graphics workstation a title frame will appear on the screen. Following this, two items will appear on the screen. One is a window from which data can be transmitted to and from the program. The window is five text lines in height and will be referred to as the text window. The other item on the right side of the screen is the Main Menu and this menu is reproduced in Fig. A.1 .

The program is mostly mouse controlled and the icon representing the mouse is a red arrow. Moving the mouse over the menu items highlights them with a different color. Depressing the right mouse button will cause execution of that menu item. A brief description of the Main Menu items follows.

#### Quit Tigg3d

When this option is selected another menu appears with two options:

**Return to Main MENU****Quit Tigg3d**

Pressing the right mouse button over either of these options will execute them and their functions obviously need no explanation.

**Write Output File**

When this option is selected, the block grid data are written to the main output file. For more detailed information on these data and file the user is referred to the **Output File Description** section of this report.

**Start/Restart Tigg3d**

When this option is selected a message is sent to the text window which says:

*input the data file name (-999 to exit)*

The mouse needs to be moved into the text window before the input file name can be typed in. Enter -999 if the reading of an input file is not desired and control will return to the Main Menu. Otherwise, after entering the input file several actions will take place. First the output file name will be assigned. It typically will be the name of the input file with a *.new* suffix attached. However, if a file with that name already exists, the user will be prompted for another name for the output file. After that, routines to generate the meridional block grid boundaries will be called and a graphics window will be opened up with a plot of the grid block boundaries in it. The workstation screen at this point will look something like Fig. A.2 .

## System Call

This option allows a *UNIX* system command to be executed from within the program. Move the mouse in the text window and enter the command. When the command has finished execution control will automatically be transferred to the Main Menu.

The other options on the Main Menu have their own menus with many options so explanation of these options will be taken up in the following sections.

### A.7.3 Edit Tools Option/Menu

When this option is selected from the Main Menu, another menu with several options is displayed and a representation of that menu is seen in Fig. A.14. The options for this menu are described below.

#### Exit Edit Tools

This option returns control of the program to the Main Menu.

## ZOOM/TRANSLATE

This option manipulates the plot in the graphics window. The operation is very similar in control to the **PLOT3D** image manipulations with the mouse. For this operation to execute the mouse must be moved into the graphics window after the menu item has been selected.

One can magnify the plot by pressing the middle mouse button and, while continuing to hold the button down, move the mouse down on the screen. The plot will

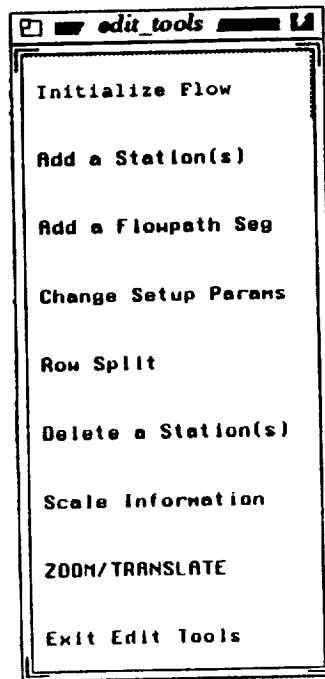


Figure A.14: Edit tools menu graphic

continue to be magnified at a certain rate until the middle mouse button is released. Similarly, one can reduce the plot by moving the mouse up on the screen while holding down the middle button.

This rate of magnification or reduction can be controlled in two ways. One control is how far the mouse moves (either up or down) while holding the middle button down (e.g., small distance for small rate, large distance for large rate). The other control is the arrow keys on the workstation keyboard. You can increase the rate by pressing the up arrow key (once or repeatedly) before the middle mouse button is held down. Similarly, the down arrow key reduces the rate of magnification.

To translate the plot the right mouse button is pressed and the mouse is moved in a direction while the button is held down. The plot will move in the direction that



the mouse moved at a certain rate until the right mouse button is released. The rate of translation is controlled the same way that the rate of magnification or reduction is controlled.

When the plot is moved and magnified in the desired way, the left mouse button can be pressed while the mouse is in the graphics window. This action returns control of the program to the menu from which the **ZOOM/TRANSLATE** option was selected.

### **Scale Information**

This option scales the grid size. After the option is selected the mouse is moved into the text window and a scale factor is entered. After entering the factor the grid block structure is resized and control is returned to the Edit Tools Menu. Since the grid ultimately produced is in non-dimensional form, this option is not usefull but is still retained from the **TIGGERC** code.

### **Delete Station(s)**

An entire set of stations from the hub to the outer boundary can be removed if this option is selected. First a station selector graphic is produced on the screen like the one represented in Fig. A.15. If the mouse is moved into one of the station selector areas, the corresponding station on the plot in the graphics window is highlighted in blue. Pressing the right mouse deletes the highlighted stations. To exit this option, press the left mouse button while the mouse is in the station selector area. This option is not recommended as it might produce unpredictable results from other sections of the program.

STATION TO DELETE
STATION # 1
STATION # 2/3
STATION # 4/5
STATION # 6/7
STATION # 8/9
STATION # 10/11
STATION # 12/13
STATION # 14/15
STATION # 16

Figure A.15: Delete station(s) selector graphic

### **Row Split**

This option is inactive. No action is taken when this option is selected.

### **Change Setup Params**

Selecting this option displays a window with the setup parameter listed. A parameter can be changed by moving the mouse over the data field where the value is displayed and pressing the right mouse button. Enter the new value and hit the return key. Save the changes or exit the display by pressing the right mouse button while the mouse is over the designated area.

### **Add a Flowpath Seg**

This option adds a flowpath segment to a designated block from the hub to the outer boundary. This option, while functional, is not supported in this manual.

### **Add a Station(s)**

An entire set of stations from the hub to the outer boundary can be added if this option is selected. First a selector graphic is produced on the screen to represent the block segments of the grid like the station selector graphic shown earlier. When a block is selected by pressing the right mouse button on a selector area, the lowest flowpath segment in the block section is highlighted. Moving the mouse into the graphics widow, a point on the highlighted flowpath can be selected by pressing the right mouse button while the mouse is over the flowpath point. This will form a lower station bound and then the next higher flowpath will be highlighted. Station

points are selected in a similar way for the entire height of the grid. This option is not generally recommended.

### **Initialize Flow**

No action is taken when this option is selected.

#### **A.7.4 Station Menu**

When this option is selected from the Main Menu, another menu with several options is displayed and a representation of that menu is seen in Fig. A.16. The options for this menu are described below.

### **Exit Station Menu**

This option returns control of the program to the Main Menu.

### **ZOOM/TRANSLATE**

This option manipulates the plot in the graphics window. The operation is very similar in control to the **PLOT3D** image manipulations with the mouse. For this operation to execute the mouse must be moved into the graphics window after the menu item has been selected.

One can magnify the plot by pressing the middle mouse button and, while continuing to hold the button down, move the mouse down on the screen. The plot will continued to be magnified at a certain rate until the middle mouse button is released.

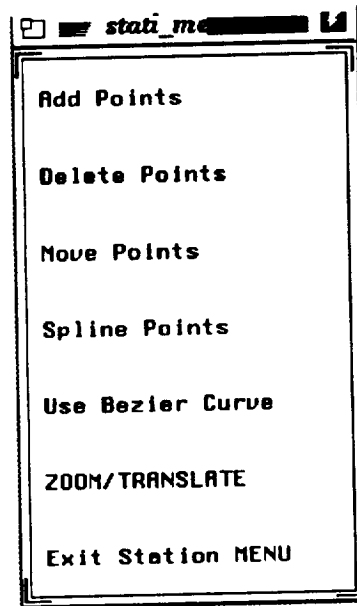


Figure A.16: Station menu graphic

Similarly one can reduce the plot by moving the mouse up on the screen while holding down the middle button.

This rate of magnification or reduction can be controlled in two ways. One control is how far the mouse moves (either up or down) while holding the middle button down (e.g., small distance for small rate, large distance for large rate). The other control is the arrow keys on the workstation keyboard. You can increase the rate by pressing the up arrow key (once or repeatedly) before the middle mouse button is held down. Similarly the down arrow key reduces the rate of magnification.

To translate the plot the right mouse button is pressed and the mouse is moved in a direction while the button is held down. The plot will move in the direction that the mouse moved at a certain rate until the right mouse button is released. The rate

of translation is controlled the same way that the rate of magnification or reduction is controlled.

When the plot is moved and magnified in the desired way, the left mouse button can be pressed while the mouse is in the graphics window. This action returns control of the program to the menu from which the **ZOOM/TRANSLATE** option was selected.

### Use Bezier Curve

This option redefines a station or segment of a station with a set of points on a bezier curve.

When this option is selected, a station selector graphic , a representation of which is shown in Fig. A.17, is produced on the screen. A series of stations are represented by block areas in the selector graphic. By moving the mouse into one of the areas, the corresponding station in the graphic window is highlighted in blue. If there are more stations in the grid than can be represented by the selector graphic, then selector control boxes like the ones shown in Fig. A.17 will be displayed as well. By pressing the right mouse button while the mouse is over the control boxes, the user can display on the station selector any station in the grid.

The user selects a station by pressing the right mouse button while the mouse is in the desired selector area. If no station is desired, then pressing the left mouse button while the mouse is in the selector area causes the program to exit back to the Station Menu.

Once a station is chosen, the selector disappears and a slider bar appears indicating the number of points that will make up the bezier curve. At this time, the

PAGE UP		SCROLL UP
PAGE DOWN		SCROLL DOWN

SELECT STATION
STATION # 1
STATION # 2/3
STATION # 4/5
STATION # 6/7
STATION # 8/9
STATION # 10/11
STATION # 12/13
STATION # 14/15
STATION # 16
STATION # 17

Figure A.17: Station selector graphic

user must select the lower and upper station bound over which the bezier curve will be produced. Any points on the station can form the bezier bounds. To select a point, the mouse must be over the point and this will be indicated by the point being highlighted from red to blue. Pressing the right mouse button selects the point.

When the lower and upper bounds have been chosen, a black pointer which follows the mouse will appear. This pointer sets the two curve control points. The curve control points set the slope and strength of the curve at the two end points. Figure A.18 shows an example of a bezier curve setup. The figure shows the two chosen station bounds and the desired locations for the curve control points. By pressing the right mouse button the control point for the lower station bound is selected. The mouse should be moved to the second control point location (for the upper bound) and the right mouse button pressed again. A bezier curve is then produced and Fig. A.19 shows the resulting curve for the example.

Once the curve has been established, the control points can be moved anywhere on the graphics window, thus redefining the curve. This is done by pressing the right mouse button while the mouse is over the control point and moving the mouse while continuing to hold down the right button. The number of points defining the curve can also be changed by moving the mouse over the slide bar indicator, pressing the right mouse button, and moving the indicator to a new bar position by moving the mouse and holding the right button down. Pressing the left mouse button exits the bezier curve operation and returns to the Station Menu.

## Spline Points



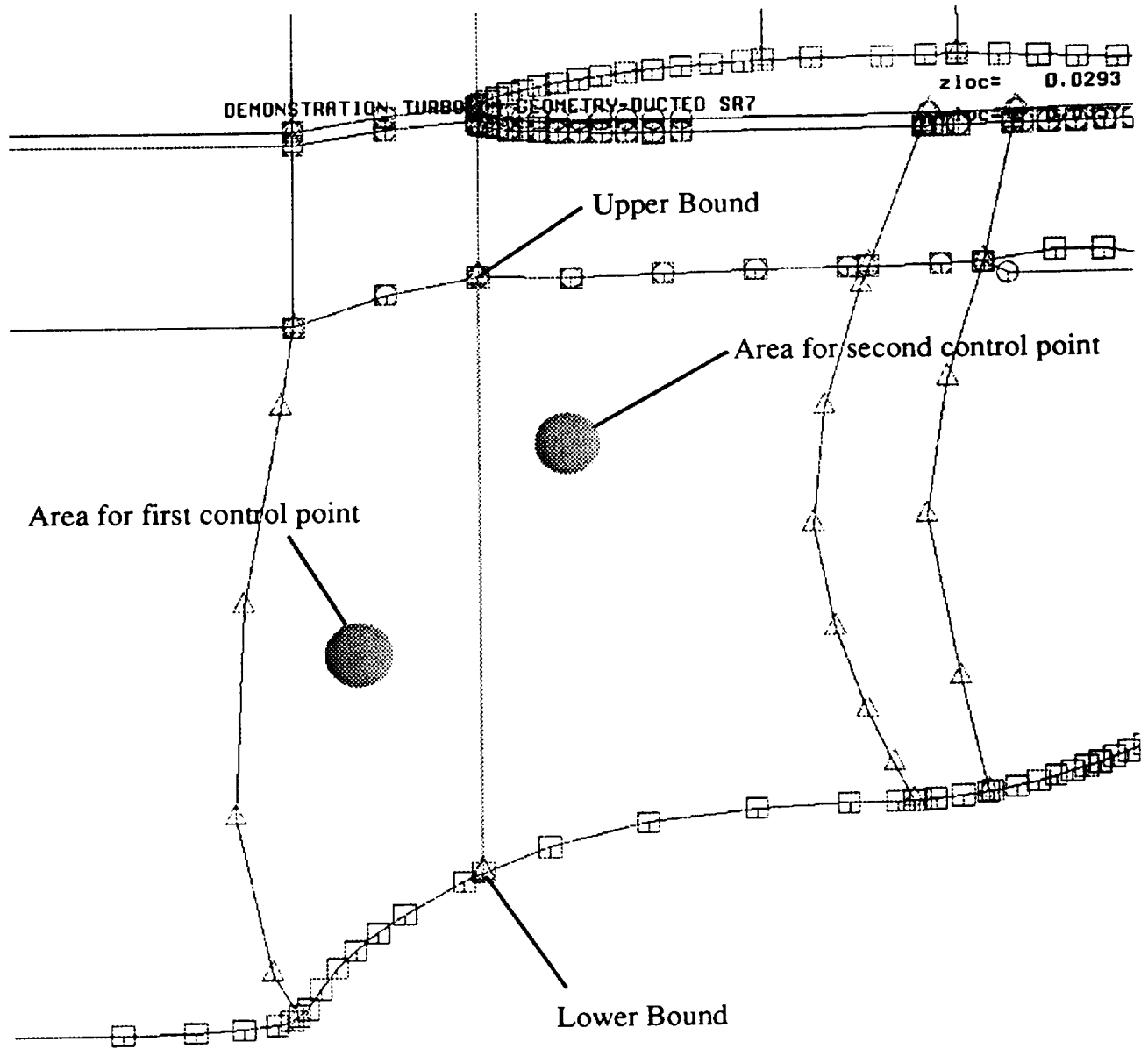


Figure A.18: Bezier curve setup

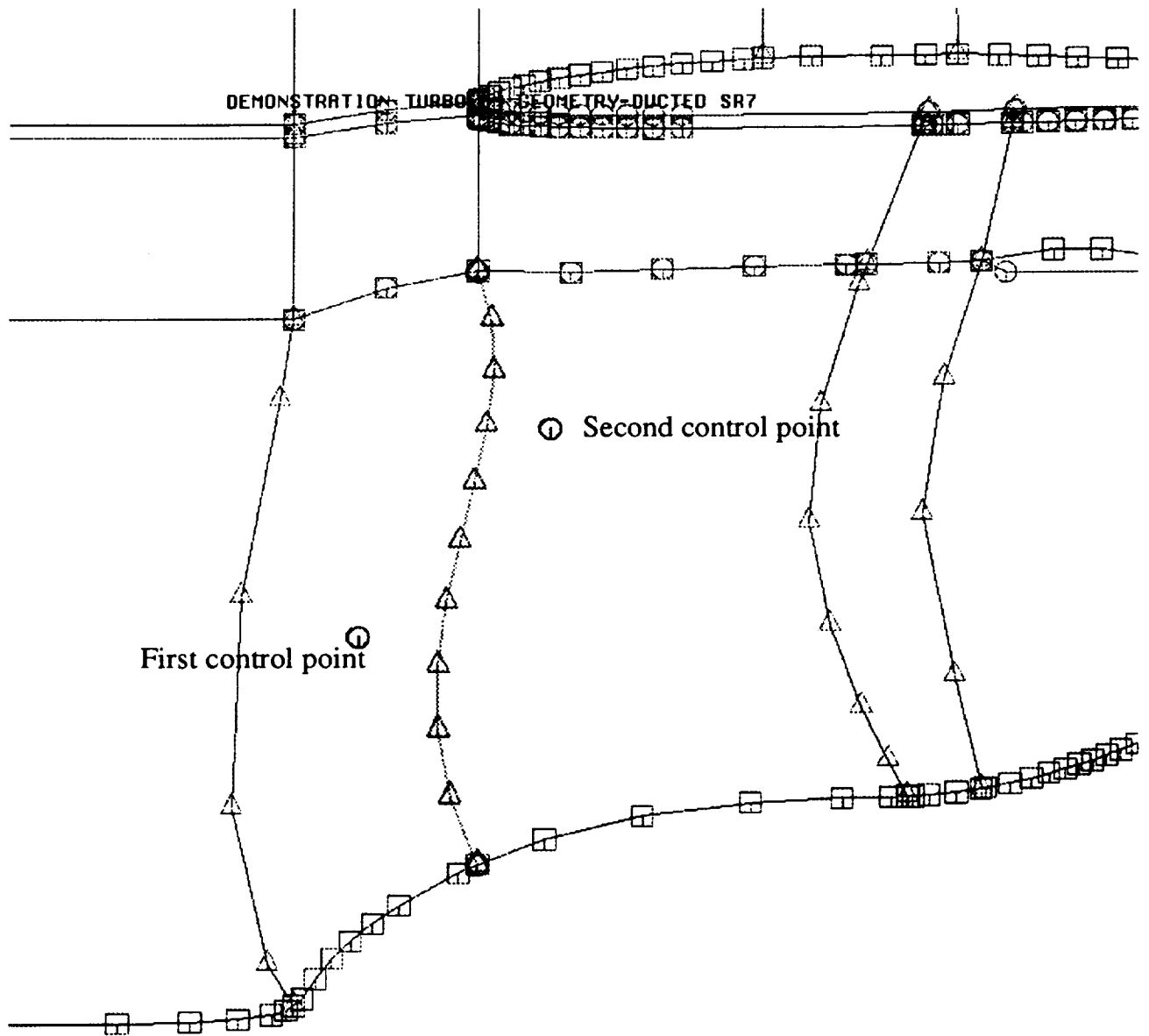


Figure A.19: Bezier curve on a station boundary

This option redefines a station or segment of a station with a set of splined points. When this option is selected, a station selector graphic, a representation of which is shown in Fig. A.17, is produced on the screen. The user is referred to the **Use Bezier Curve** operation described previously in this section for information on the station selector.

Once a station is chosen, the selector disappears and a slider bar appears indicating the number of points that will make up the new splined curve. At this time the user must select the lower and upper station bound over which the splined curve will be produced. Any points on the station can form the bounds. To select a point the mouse must be over the point and this will be indicated by the point being highlighted from red to blue. Pressing the right mouse button selects the point. Figure A.20 shows an example of a spline curve setup.

Once the bounds are chosen, these and any points in between form the basis of the spline curve. The curve is then produced and the example is shown in Fig. A.21.

The number of points defining the curve can be changed by moving the mouse over the slide bar indicator, pressing the right mouse button, and moving the indicator to a new bar position by moving the mouse and holding the right button down. Pressing the left mouse button exits the spline operation, saves the new points, and returns to the Station Menu. Pressing the center mouse button exits the spline operation without saving the changes.

### Move Points

This option moves points on a selected station boundary.

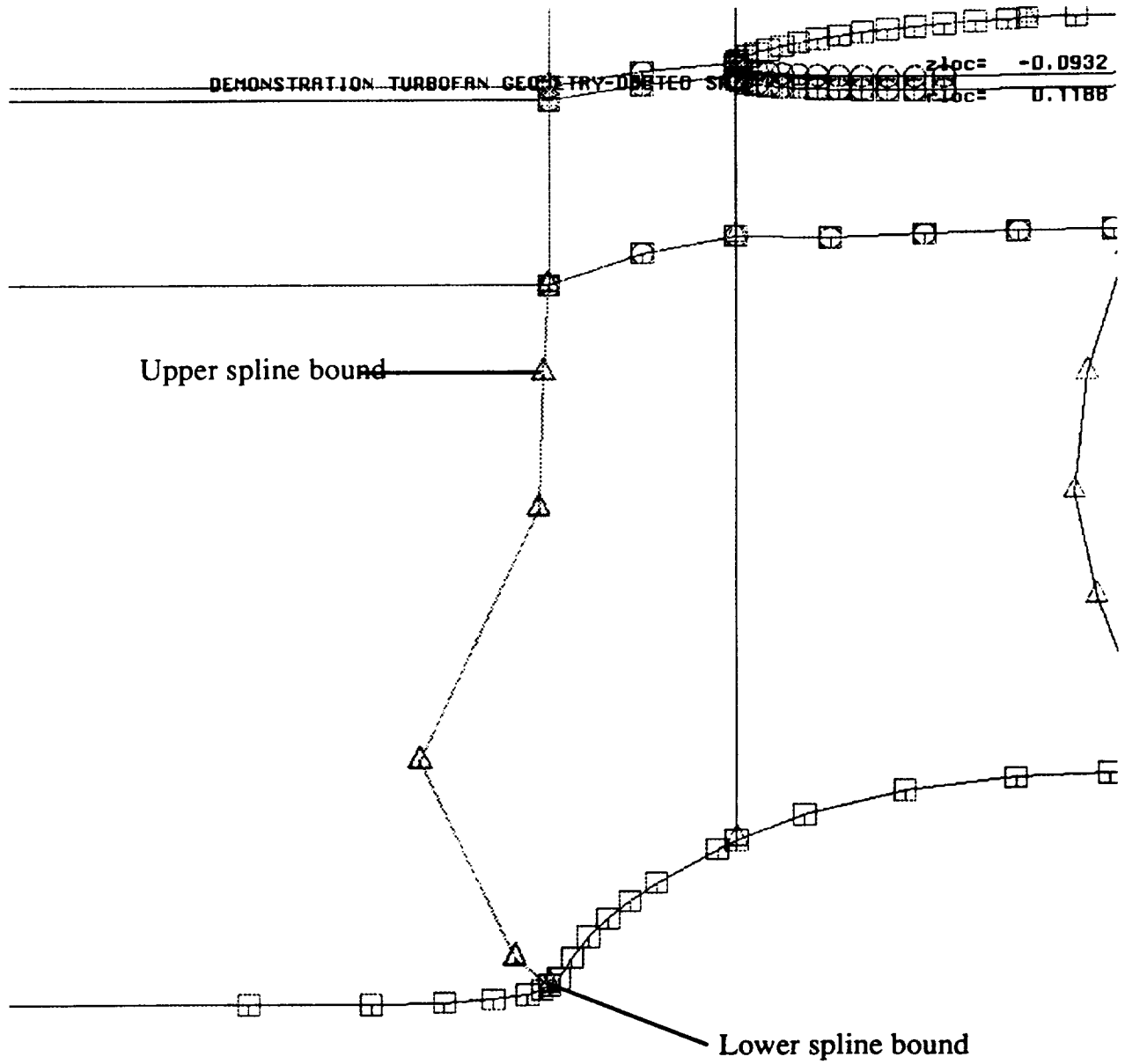


Figure A.20: Spline interpolation setup

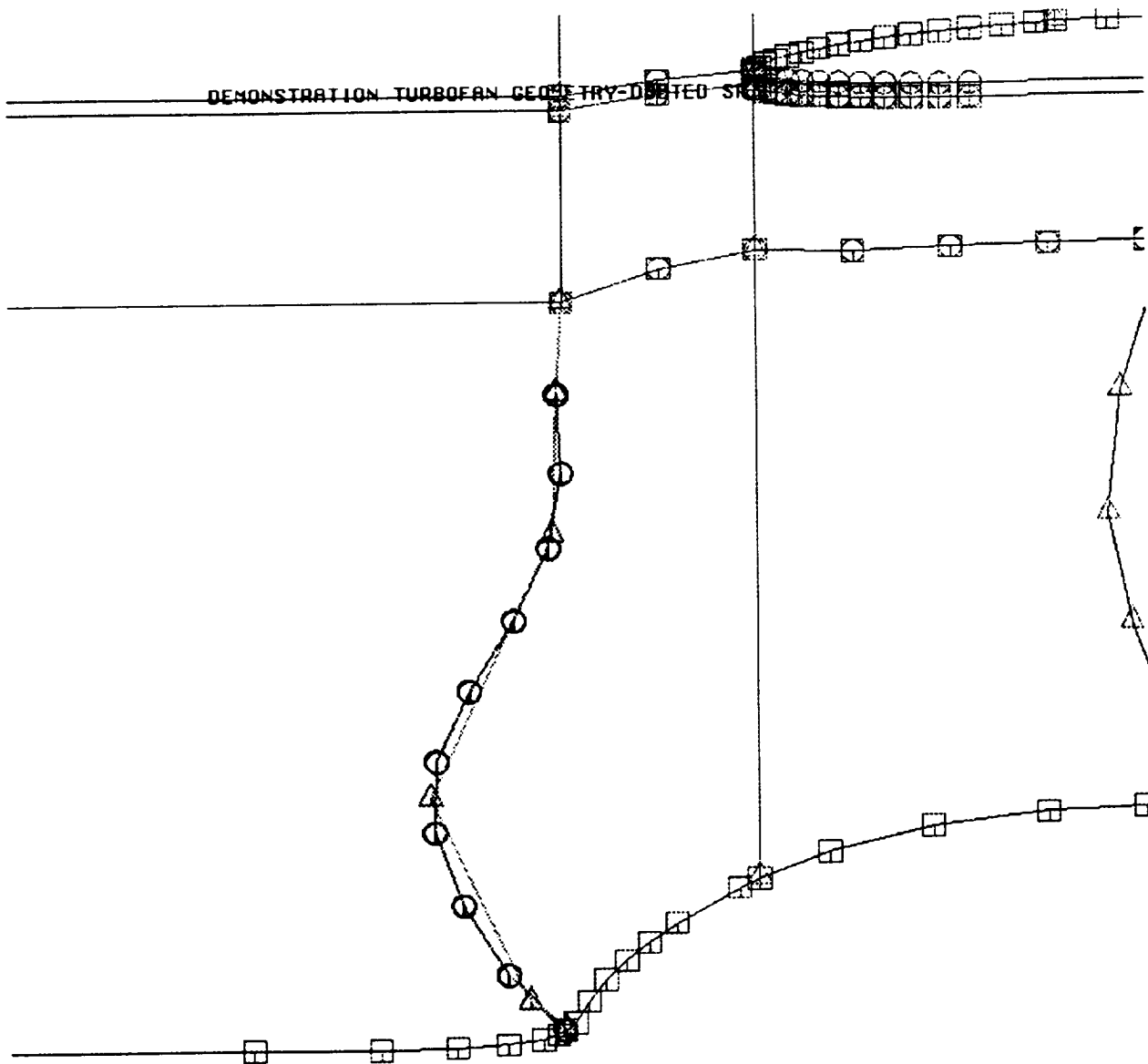


Figure A.21: Spline interpolation on a station boundary

When this option is selected, a station selector graphic, a representation of which is shown in Fig. A.17, is produced on the screen. The user is referred to the **Use Bezier Curve** operation described previously in this section for information on the station selector.

Once the station is selected a point is moved by bringing the mouse to the desired point (highlighted in blue), pressing the right mouse button and holding it down while moving the mouse. The point will then move in the direction that the mouse moves at a certain rate until the button is released. This rate of movement can be controlled in two ways. One control is how far the mouse moves while holding the right button down (e.g., small distance for small rate, large distance for large rate). The other control is the arrow keys on the workstation keyboard. You can increase the rate by pressing the up arrow key (once or repeatedly) after the right mouse button is held down. Similarly the down arrow key reduces the rate of magnification. Pressing the left mouse exits out of the station move operation.

### **Delete Points**

This option delete points on a selected station boundary.

When this option is selected, a station selector graphic, a representation of which is shown in Fig. A.17, is produced on the screen. The user is referred to the **Use Bezier Curve** operation described previously in this section for information on the station selector.

Once the station is selected a point is deleted by bringing the mouse to the desired point (highlighted in blue), and pressing the right mouse button. Pressing the left mouse exits out of the delete option.

## **Add Points**

This option adds points to a selected station boundary.

When this option is selected, a station selector graphic, a representation of which is shown in Fig. A.17, is produced on the screen. The user is referred to the **Use Bezier Curve** operation described previously in this section for information on the station selector.

Once the station is chosen the user must then select the lower and upper station bounds between which the additional points will be placed. Any points on the station can form the bounds. To select a point, the mouse must be over the point and this will be indicated by the point being highlighted from red to blue. Pressing the right mouse button selects the point. After the bounds have been selected, a black pointer will appear and follow the mouse. By pressing the right mouse button, a red symbol will be placed and that will be the position of the first added point. Other points can be added by repositioning the mouse and again pressing the right mouse button. The new station segment will start at the lower bound and connect the first, second, and so forth, added points and end at the upper bound. Pressing the left mouse exits out of the option and saves the new station points.

### **A.7.5 Flowpath Menu**

When this option is selected from the Main Menu another menu with several options is displayed and a representation of that menu is seen in Fig. A.22. The options for this menu are described below.

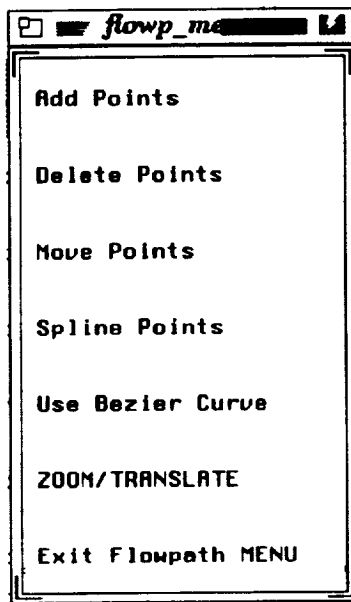


Figure A.22: Flowpath menu graphic

### **Exit flowpath Menu**

This option returns control of the program to the Main Menu.

### **ZOOM/TRANSLATE**

This option manipulates the plot in the graphics window. The operation is very similar in control to the **PLOT3D** image manipulations with the mouse. For a detailed description of this menu option the user is referred to the **ZOOM/TRANSLATE** operation described previously in **Station MENU** section.



### Use Bezier Curve

This option redefines a flowpath or segment of a flowpath with a set of points on a bezier curve.

When this option is selected, a flowpath selector graphic, a representation of which is shown in Fig. A.23, is produced on the screen. A series of flowpath boundaries are represented by block areas in the selector graphic. By moving the mouse into one of the areas, the corresponding hub or casing flowpath in the graphic window is highlighted in green. If there are more flowpaths in the grid than can be represented by the selector graphic, then selector control boxes like the ones shown in Fig. A.23 will be displayed as well. By pressing the right mouse button while the mouse is over the control boxes, the user can display on the flowpath selector any hub or casing boundary in the grid.

The user selects a flowpath by pressing the right mouse button while the mouse is in the desired selector area. If no flowpath is desired, then pressing the left mouse button while the mouse is in the selector area causes the program to exit back to the Flowpath Menu.

If the flowpath boundary chosen has a neighboring block boundary defined by the same points, then another window will appear over the graphic window. A question will appear in the window:

*Move the flowpaths separately?*

Almost always the appropriate answer to this question is no, and is executed by moving the mouse to the answer box and pressing the right mouse button.

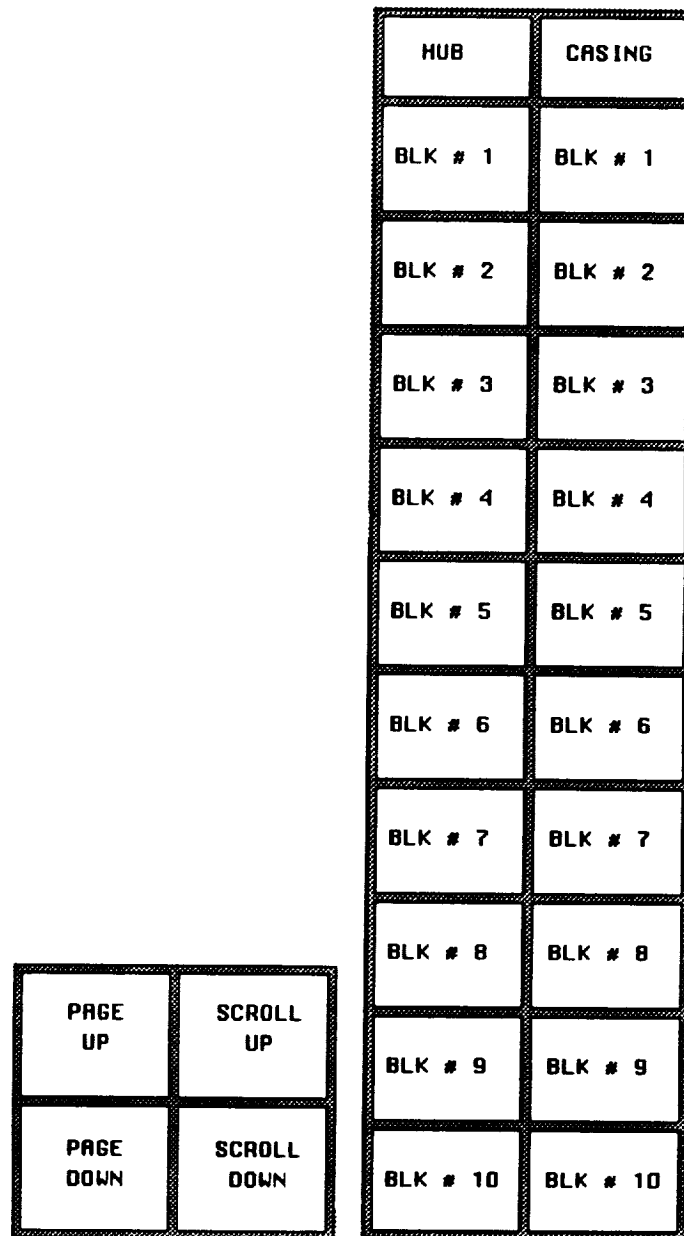


Figure A.23: Flowpath selector graphic

Once a flowpath is chosen the selector disappears and a slider bar appears indicating the number of points that will make up the bezier curve. At this time, the user must select the front and rear flowpath bound over which the bezier curve will be produced. Any points on the flowpath can form the bezier bounds. To select a point, the mouse must be over the point and this will be indicated by the point being highlighted from blue to red. Pressing the right mouse button selects the point.

When the lower and upper bounds have been chosen, a black pointer appears and follows the mouse. This pointer sets the two curve control points. The curve control points set the slope and strength of the curve at the two end points. By pressing the right mouse button the control point for the front flowpath bound is selected. The mouse should be moved to the second control point location (for the rear bound) and the right mouse button pressed again. A bezier curve is then produced for the flowpath segment.

Once the curve has been established, the control points can be moved anywhere on the graphics window, thus redefining the curve. This is done by pressing on the right mouse button while the mouse is over the control point and moving the mouse while continuing to hold down the right button. The number of points defining the curve can also be changed by moving the mouse over the slide bar indicator, pressing the right mouse button, and moving the indicator to a new bar position by moving the mouse and holding the right button down. Pressing the left mouse button exits the bezier curve operation and returns to the Flowpath Menu.

### **Spline Points**

This option redefines a flowpath or segment of a flowpath with a set of splined points.

When this option is selected, a flowpath selector graphic is produced on the screen and a representation of that is shown in Fig. A.23. The user is referred to the **Use Bezier Curve** described previously in this section for information on the flowpath selector.

If the flowpath boundary chosen has a neighboring block boundary defined by the same points, then another window will appear over the graphic window. A question will appear in the window:

*Do you want to spline on both flowpaths?*

Almost always the appropriate answer to this question is yes, and is executed by moving the mouse to the answer box and pressing the right mouse button.

Once a station is chosen, the selector disappears and a slider bar appears indicating the number of points that will make up the new splined curve. At this time the user must select the front and rear flowpath bound over which the splined curve will be produced. Any points on the flowpath can form the bounds. To select a point the mouse must be over the point and this will be indicated by the point being highlighted from blue to red. Pressing the right mouse button selects the point.

Once the bounds are chosen, these and any points in between form the basis of the spline curve and the curve is then produced. The number of points defining the curve can be changed by moving the mouse over the slide bar indicator, pressing the right mouse button, and moving the indicator to a new bar position by moving the mouse and holding the right button down. Pressing the left mouse button exits the spline operation, saves the new points, and returns to the Flowpath Menu. Pressing the center mouse button exits the spline operation without saving the changes.

## Move Points

This option moves points on a selected flowpath boundary.

When this option is selected, a flowpath selector graphic is produced on the screen and a representation of that is shown in Fig. A.23. The user is referred to the **Use Bezier Curve** operation described previously in this section for information on the flowpath selector.

If the flowpath boundary chosen has a neighboring block boundary defined by the same points, then another window will appear over the graphic window. A question will appear in the window:

*Do you want to move the congruent points separately?*

Almost always the appropriate answer to this question is no, and is executed by moving the mouse to the answer box and pressing the right mouse button.

Once the flowpath is selected, a point is moved by bringing the mouse to the desired point (highlighted in red), pressing the right mouse button and holding it down while moving the mouse. The point will then move in the direction that the mouse moves at a certain rate until the button is released. This rate of movement can be controlled in two ways. One control is how far the mouse moves while holding the right button down (e.g., small distance for small rate, large distance for large rate). The other control is the arrow keys on the workstation keyboard. You can increase the rate by pressing the up arrow key (once or repeatedly) after the right mouse button is held down. Similarly the down arrow key reduces the rate of magnification. Pressing the left mouse exits out of the move operation.

### **Delete Points**

This option delete points on a selected flowpath boundary.

When this option is selected, a flowpath selector graphic, shown in Fig. A.23, is produced on the screen. The user is referred to the **Use Bezier Curve** operation described previously in this section for information on the flowpath selector.

Once the flowpath is selected, a point is deleted by bringing the mouse to the desired point (highlighted in red), and pressing the right mouse button. Pressing the left mouse exits out of the delete option.

### **Add Points**

This option adds points to a selected station boundary.

When this option is selected, a flowpath selector graphic, shown in Fig. A.23, is produced on the screen. The user is referred to the **Use Bezier Curve** operation described previously in this section for information on the flowpath selector.

Once the flowpath is chosen, the user must then select the front and rear flowpath bound between which the additional points will be placed. Any points on the flowpath can form the bounds. To select a point, the mouse must be over the point and this will be indicated by the point being highlighted from blue to red. Pressing the right mouse button selects the point. After the bounds have been selected, a black pointer will appear and follow the mouse. By pressing the right mouse button, a red symbol will be placed and that will be the position of the first added point. Other points can be added by repositioning the mouse and again pressing the right mouse button. The new flowpath segment will start at the front bound and connect the first, second, and

so forth, added points and end at the rear bound. Pressing the left mouse exits out of the option and saves the new flowpath points.

### **A.7.6 Block Menu**

When this option is selected from the Main Menu a mesh is generated for all the meridional grid blocks and it is displayed in the graphics window. Also another menu with several options is displayed and a representation of that menu is seen in Fig. A.24. The options for this menu are described below.

#### **Exit Block Menu**

This option returns control of the program to the Main Menu.

#### **ZOOM/TRANSLATE**

This option manipulates the plot in the graphics window. The operation is very similar in control to the **PLOT3D** image manipulations with the mouse. For a detailed description of this menu option the user is referred to the **ZOOM/TRANSLATE** operation described previously in **Station MENU** section.

#### **R-Theta View/Mod**

When this option is selected from the Block Menu, another menu with several options having to do with the tangential grid manipulation is displayed. A detailed description of this portion of the program will be taken up in another section.

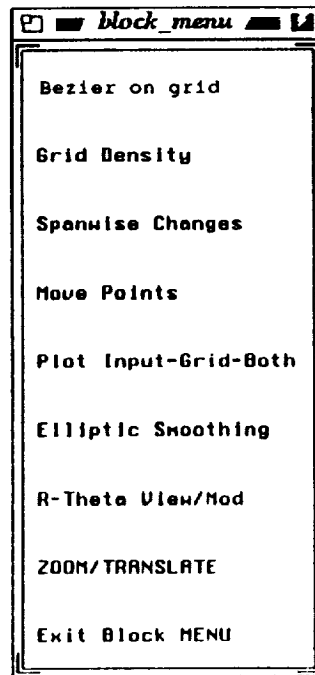


Figure A.24: Block menu graphic

### Elliptic Smoothing

This option executes the elliptic smoothing procedure on the meridional plane grid. Parameters set in the input file control the number of iterations the smoothing operator performs on the grid. When the procedure is finished, the “smoothed” grid will be plotted in the graphics window. After this operation is finished and the smooth grid is desired, the *Spanwise Changes* and *Grid Density* options should not be selected, as they would regenerate the grid with the algebraic spacing once again.

### Plot Input-Grid-Both

This option plots up either the meridional plane grid, the block boundary structure input to the mesh generator, or both the grid and the block boundary input.



The plot selection is toggled upon pressing the right mouse button while the mouse is in the menu option area.

### **Move Points**

This option moves individual points on the meridional plane grid.

When this option is selected, a grid block selector graphic is produced on the right side of the workstation screen. A series of grid blocks are represented by areas in the selector graphic. By moving the mouse into one of the areas, the corresponding block in the meridional plane grid in the graphic window is highlighted in blue. If there are more blocks in the grid than can be represented by the selector graphic, then selector control boxes will be displayed as well. By pressing the right mouse button while the mouse is over the control boxes, the user can display on the selector any block in the grid.

The user selects a block by pressing the right mouse button while the mouse is in the desired selector area. If no block is desired, then pressing the left mouse button while the mouse is in the selector area causes the program to exit back to the Block Menu.

Once the grid block is chosen, a point is moved by bringing the mouse to the desired point (highlighted by a red circle around the point), pressing the right mouse button and holding it down while moving the mouse. The point will then move in the direction that the mouse moves at a certain rate until the button is released. This rate of movement can be controlled in two ways. One control is how far the mouse moves while holding the right button down (e.g., small distance for small rate, large distance for large rate). The other control is the arrow keys on the workstation keyboard. You

can increase the rate by pressing the up arrow key (once or repeatedly) after the right mouse button is held down. Similarly the down arrow key reduces the rate of magnification. Pressing the left mouse exits out of the grid point move operation.

### Spanwise Changes

This option changes the number of spanwise grid points and the spanwise spacing of those points.

When this option is selected, a grid block row modifier graphic, shown in Fig. A.25, is produced on the screen. A series of grid block rows are represented by areas in the modifier graphic. By moving the mouse into one of the areas, the corresponding row in the meridional plane grid in the graphic window is highlighted in blue.

The number of spanwise points in the row can be changed by moving the mouse over the slide bar indicator, pressing the right mouse button, and moving the indicator to a new bar position by moving the mouse and holding the right button down.

The spanwise spacing is controlled by a minimum spacing distance from the top (or shroud) of the row boundary  $dxs$ , and a minimum distance from the bottom (or hub) of the row  $dxh$ . These factors set a hyperbolic tangent spacing distribution for the radial grid segment lengths. A geometric spacing distribution can also be produced if it is chosen in the Edit Tools Menu. Then the factors  $srs$  and  $srh$  will set the maximum ratio of neighboring grid segment lengths radially from the shroud and hub boundaries, respectively. To change the spanwise spacing distribution, the mouse is moved into the area in the modifier containing the spacing factor and the right mouse button pressed. This action blanks out the old number and the user enters the new number from the keyboard and presses the return key.

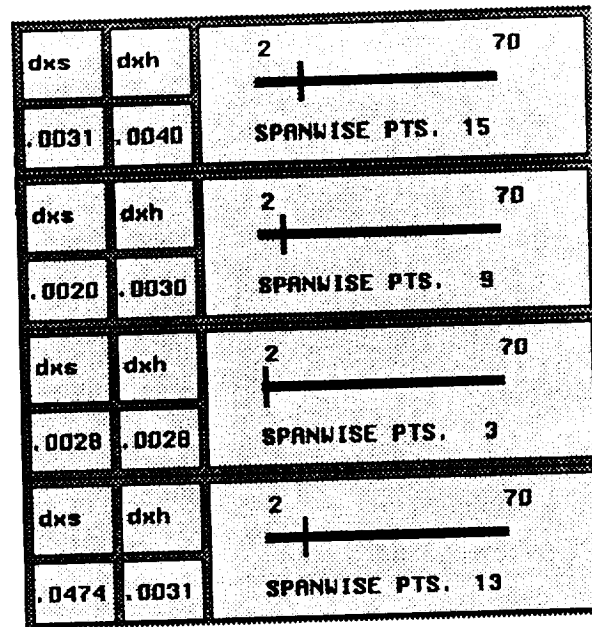


Figure A.25: Grid block row modifier graphic

To leave the span change operation, the left mouse button is pressed while the mouse is in the modifier area.

### Grid Density

This option changes the number of axial grid points and the spacing of those points.

When this option is selected, a grid block modifier graphic, shown in Fig. A.26, is produced on the screen. A series of grid blocks are represented by areas in the modifier graphic. By moving the mouse into one of the areas, the corresponding block of the meridional plane grid in the graphic window is highlighted in blue.

The number of axial points in the block can be changed by moving the mouse over the slide bar indicator, pressing the right mouse button, and moving the indicator

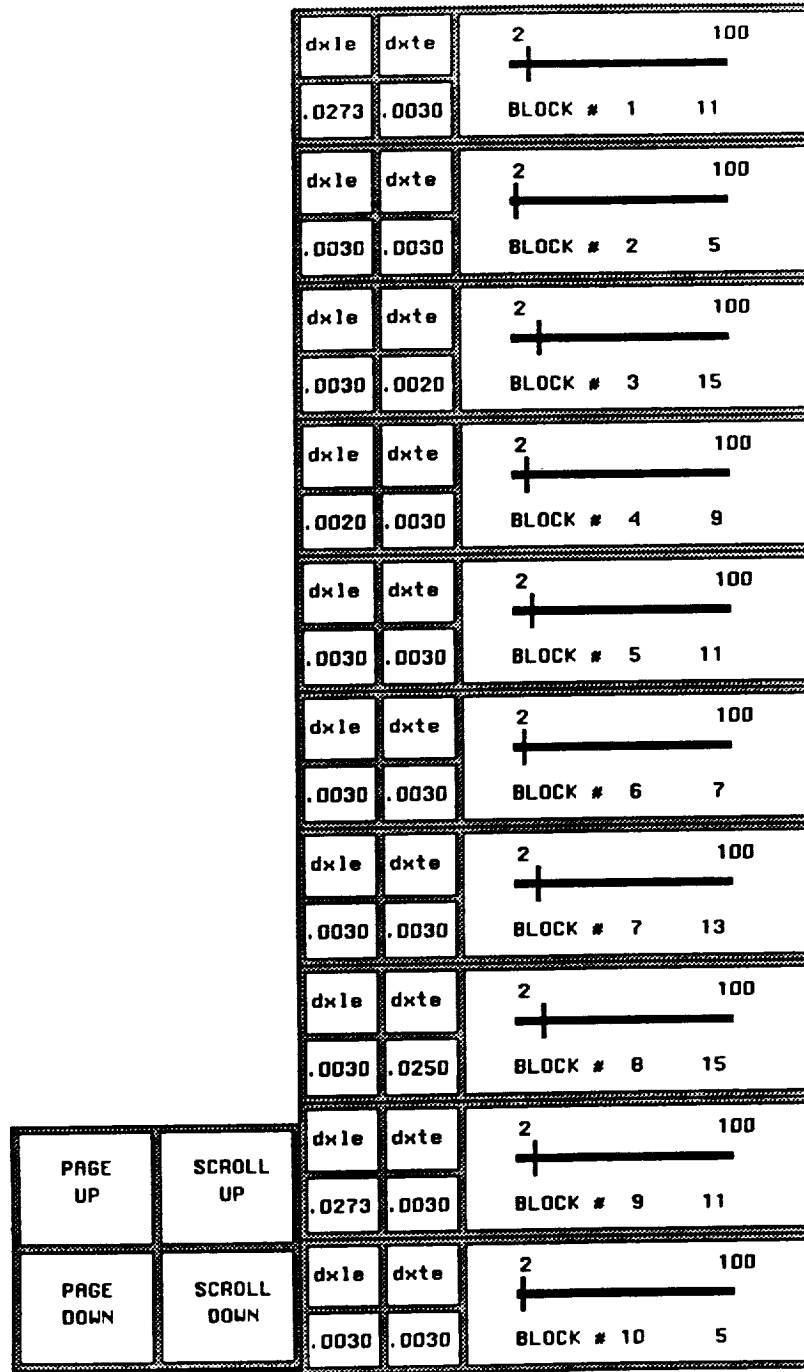


Figure A.26: Grid block modifier graphic

to a new bar position by moving the mouse and holding the right button down.

The spacing is controlled by a minimum spacing distance from the front of the grid block boundary  $dxle$ , and a minimum distance from the rear of the block  $dxte$ . These factors set a hyperbolic tangent spacing distribution. A geometric spacing distribution can also be produced if it is chosen in the Edit Tools Menu. Then the factors  $sle$  and  $ste$  will set the maximum ratio of neighboring grid segment lengths from the leading edge and trailing edge, respectively. To change the spacing distribution, the mouse is moved into the area in the modifier containing the spacing factor and the right mouse button pressed. This action blanks out the old number and the user enters the new number from the keyboard and presses the return key.

To leave the grid density operation, the left mouse button is pressed while the mouse is in the modifier area.

### Bezier on Grid

This option conforms grid points in a spanwise or axial grid line to a bezier curve.

When this option is selected, a grid block selector graphic is produced on the screen and a description of the use of this selector can be found earlier in this section under the **Move Points** option.

After a grid block is chosen, the user must select the axial or spanwise bounds over which the bezier curve will be produced. To select a point the mouse must be over the point and this will be indicated by the point being encircled by a red symbol. Pressing the right mouse button selects the point.

When the bounds have been chosen, a black pointer appears and follows the mouse. This pointer sets the two curve control points. The curve control points set

the slope and strength of the curve at the two end points. By pressing the right mouse button the control point for the first station bound is selected. The mouse should be moved to the second control point location and the right mouse button pressed again. The grid points are then aligned along the bezier curve.

Once the curve has been established the control points can be moved anywhere on the graphics window, thus redefining the curve. This is done by pressing on the right mouse button while the mouse is over the control point and moving the mouse while continuing to hold down the right button. Pressing the left mouse button exits the bezier curve operation and returns to the Block Menu.

#### **A.7.7 R-Theta View/Mod**

When this option is selected from the Block Menu, the tangential grid is generated and the grid at the first axial position is plotted on the graphics window. If there is more than one blade row, then the user is prompted to enter which blade row the tangential grid generator will accurately model. Also, another menu with several options is displayed and the options for this menu are described below.

#### **Select Block/Slice**

This menu option plots a particular axial grid section into the graphics window. A message appears in the text window like the one below

*Enter the base block and the axial section of that block*

When the user enters a number for a block on the first block row and the axial index for that block separated by a space, that grid section is plotted in the graphics window. Control is automatically returned to the Theta Menu.

### **Zoom/Translate**

This option manipulates the plot in the graphics window. The operation is very similar in control to the **PLOT3D** image manipulations with the mouse. For a detailed description of this menu option the user is referred to the **ZOOM/TRANSLATE** operation described previously in **Station MENU** section.

### **Theta Parameters**

This option changes the number of tangential grid points and the spacing of those points. When this option is selected, a tangential modifier graphic, shown in Fig. A.27, is produced on the screen.

The number of tangential points in the grid can be changed by moving the mouse over the slide bar indicator, pressing the right mouse button, and moving the indicator to a new bar position by moving the mouse and holding the right button down.

The tangential spacing is controlled by a minimum spacing distance from the top blade surface  $dt1$ . To change the spacing distribution, the mouse is moved into the area in the modifier containing the spacing factor and the right mouse button pressed. This action blanks out the old number and the user enters the new number from the keyboard and presses the return key.

To leave this operation the left mouse button is pressed while the mouse is in the modifier area.

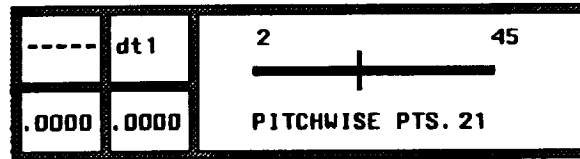


Figure A.27: Tangential grid modifier graphic

### Write PLOT3D Grid

This option writes out the three-dimensional grid.

The multiple-block grid is synthesized into one grid and is written out in a format compatible with a **PLOT3D** multiple grid binary file.

### Exit Theta MENU

This option returns control of the program to the Block Menu.



## A.8 *TIGG3DB*: BATCH OPERATION

This section outlines the operations of the nongraphic-interactive or “batch” version of the **TIGG3D** grid generator. This code is basically the **TIGG3D** grid generation code without any graphics calls or routines. A brief description of its operation follows.

The program can be run with some interaction with the user or in a fully batch orientation at the user’s discretion. If the **TIGG3DB** code is run semi-interactively, at most two times the program prompts the user during the course of execution with the statements:

**Enter the input data file name**

and

**Enter the blade row theta grid to be generated**

The second prompt only occurs if there is more than one blade row in the geometry. The user can answer those questions interactively or enter those answers in a file and run the program with that file as an input.

For example, there is an input file for a turbofan with two blade rows called *fan.inp*. The user can generate a grid by creating a file called

*batch.dat*

in which the first two lines look like the following

```
line ....+....1....+....2....+....3....
```

```
0001 fan.inp
```

```
0002 1
```

Those two lines will instruct the program to read the input data from the file *fan.inp* and to generate a tangential grid oriented about the first blade row. The program should be executed with the statement:

```
tigg3db < batch.dat
```

This concludes the user's manual for the **TIGG3D** program.



## Report Documentation Page

1. Report No. NASA CR-187129	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems, Task III-Advanced Fan Section Grid Generator Final Report & Computer Program User's Manual		5. Report Date September 1991
7. Author(s) Andrew J. Crook Robert A. Delaney		6. Performing Organization Code
9. Performing Organization Name and Address Allison Gas Turbine Division of General Motors P.O. Box 420, SC T14A Indianapolis, IN 46206-0420		8. Performing Organization Report No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lewis Research Center 21000 Brookpark Road Cleveland, OH 44135-3191		10. Work Unit No. 535-03-10
15. Supplementary Notes Prepared in cooperation with NASA Project Manager, C.J. Miller, NASA Lewis Research Center, Cleveland, OH		11. Contract or Grant No. NAS3-25270
16. Abstract: This document is a final report on a procedure for generating three-dimensional grids for advanced turbofan engine fan section geometries. The procedure constructs a discrete mesh about engine sections containing the fan stage, an arbitrary number of axisymmetric radial flow splitters, a booster stage, and a bifurcated core/bypass flow duct with guide vanes. The mesh is an H-type grid system, the points being distributed with a transfinite interpolation scheme with axial and radial spacing being user specified. Elliptic smoothing of the grid in the meridional plane is a post-process option. The grid generation scheme is consistent with aerodynamic analyses utilizing the average-passage equation system developed by Dr. John Adamczyk of NASA Lewis. This flow solution scheme requires a series of blade specific grids each having a common axisymmetric mesh, but varying in the circumferential direction according the geometry of the specific blade row.		13. Type of Report and Period Covered
17. Key Words (Suggested by Author(s)) Grid, Mesh, Mesh Generator, Turbofan, Propfan, Ducted, NPSS		14. Sponsoring Agency Code
18. Distribution Statement Unclassified-Unlimited		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 115
		22. Price





